# Analysis and design of a random impulse signal generator suitable for power system identification

by

Alford Mbongeni Sibanda
18644570




*Report submitted in partial fulfilment of the requirements of the module Project (E) 448 for the degree Baccalaureus in Engineering in the Department of Electrical and Electronic Engineering at the University of Stellenbosch*

Supervisor: Mr. F. M. Mwaniki

November 2018

# Declaration

**Plagiaatverklaring / Plagiarism *Declaration***

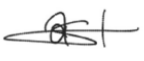
1. Plagiaat is die oorneem en gebruik van die idees, materiaal en ander intellektuele eiendom van ander persone asof dit jou eie werk is.
   *Plagiarism is the use of ideas, material and other intellectual property of another's work and to present is as my own*

2. Ek erken dat die pleeg van plagiaat 'n strafbare oortreding is aangesien dit 'n vorm van diefstal is.
   *I agree that plagiarism is a punishable offence because it constitutes theft.*

3. Ek verstaan ook dat direkte vertalings plagiaat is.
   *I also understand that direct translations are plagiarism*

4. Dienooreenkomstig is alle aanhalings en bydraes vanuit enige bron (ingesluit die internet) volledig verwys (erken). Ek erken dat die woordelikse aanhaal van teks sonder aanhalingstekens (selfs al word die bron volledig erken) plagiaat is.
   *Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism*

5. Ek verklaar dat die werk in hierdie skryfstuk vervat, behalwe waar anders aangedui, my eie oorspronklike werk is en dat ek dit nie vantevore in die geheel of gedeeltelik ingehandig het vir bepunting in hierdie module/werkstuk of 'n ander module/werkstuk nie.
   *I declare that the work contained in this assignment, except where otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.*

| **18644570** |  |
|---|---|
| Studentenommer/ Student Number | Handtekening/ Signature |
| **AM Sibanda** | **05 November 2018** |
| Voorletter en van/ Initials and surname | Datum/ Date |

# Abstract


## Analysis and design of a random impulse signal generator suitable for power system identification

A. M. Sibanda

*Department of Electrical and Electronic Engineering,*
*University of Stellenbosch,*
*Private Bag X1, Matieland 7602, South Africa.*

Thesis: BEng (Elec)

November 2018

The aim of the project was to design an excitation signal suitable for power system identification. Various excitation signals were studied in order to investigate their suitability. The designed signal was a combination of random binary sequences and current impulses. The signal was designed from first principles and then simulated in MATLAB.

A physical circuit was then built in order to excite an off-line power system for the purpose of system identification. The circuit control was achieved through serial communication to a micro-controller connected to the switches. The measured data was compared to the simulation data in order to verify the functionality of the signal generator. The signal generator was limited by the accuracy of the available lab test equipment.

The obtained results indicate that the designed signal and circuit can perturb a power system for the purpose of system identification. It is recommended that future designs should include snubbers when operated at higher voltages and that test equipment should be rated for wideband spectrum ($> 100kHz$).

# Uittreksel


## Ontleding en ontwerp van 'n impulsgenerator wat geskik is vir kragstelselidentifikasie

*("Analysis and design of a random impulse signal generator suitable for power system identification")*

A. M. Sibanda

*Departement Elektriese en Elektroniese Ingenieurswese,*
*Universiteit van Stellenbosch,*
*Privaatsak X1, Matieland 7602, Suid Afrika.*

Tesis: BIng (EE)

November 2018

Die doel van die projek was om 'n opwekking sein te ontwerp wat geskik is vir kragstelselidentifikasie. Verskeie eksitasie seine is bestudeer om hul geskiktheid te ondersoek. Die ontwerpte sein was 'n kombinasie van willekeurige binere reekse en huidige impulse. Die sein is van eerste beginsels ontwerp en dan in MATLAB gesimuleer.

'N Fisiese stroombaan is dan gebou om 'n off-line kragstelsel op te wek vir die doel van stelselidentifikasie. Die stroombaanbeheer is bereik deur middel van seriele kommunikasie na 'n mikrobeheerder wat aan die skakelaars gekoppel is. Die gemete data is vergelyk met die simulasie data om die funksionaliteit van die seingenerator te verifieer. Die seinopwekker is beperk deur die akkuraatheid van die beskikbare laboratoriumtoets toerusting.

Die resultate wat verkry is, dui daarop dat die ontwerp sein en stroombaan 'n kragstelsel kan stuit vir die doel van stelselidentifikasie. Dit word aanbeveel dat toekomstige ontwerpe snubbers insluit wanneer dit by hoer spannings gebruik word en dat toetstoerusting vir breedbandspektrum ($> 100kHz$) aangewys moet word.

# Acknowledgements

First of all, I give all thanks to God for continually strengthening me and guiding my path. I would like to express my sincere gratitude to the following

- Mr Mwaniki, for his tireless support and guidance throughout the project.
- The machines lab department for provision of test equipment.
- Jenny Martin and Lezel Jansen for ordering my components on time.
- Mr Van Eeden for local prototypes PCBs
- JLCPCB, for their high quality PCBs and fast turn around times.
- Friends and family for bearing with me throughout the project.

# Contents

# List of Figures

# List of Tables



# Listings

# Nomenclature

**Abbreviations**

| | |
|---|---|
| AC | Alternating Current |
| DC | Direct Current |
| DFT | Discrete Fourier Transform |
| DG | Distributed Generation |
| DUT | Device Under Test |
| FFT | Fast Fourier Transform |
| FRA | Frequecy-Response Analysis |
| GUI | Graphical User Interface |
| Hz | Hertz |
| IFRA | Impulse Frequecy-Response Analysis |
| IGBT | Insulated-Gate Bipolar Transistor |
| LFSR | Linear Feedback Shift Register |
| LTI | Linear Time-Invariant |
| LV | Low-voltage |
| MLBS | Maximum-Length Binary Sequence |
| ODE | Ordinary Differential Equation |
| PRBS | Pseudo-Random Binary Sequence |
| PCB | Printed Circuit Board |
| PRIS | Pseudo-Random Impulse Sequence |
| PSD | Power Spectral Density |
| RMS | Root Mean Square |
| SNR | Signal-to-Noise Ratio |
| SFRA | Sweep Frequecy-Response Analysis |
| XOR | Exclusive Or |

# Chapter 1

# Introduction

## 1.1 Background

The advent of renewable energy has changed the way energy is viewed globally. The rapid rise of renewable energy in recent decades has seen it transition from off-grid electricity supply for private use to being integrated into the traditional power grid [1]. The integration of renewable energy sources to the grid can affect the stability of the current injected into the system [2]. This is attributed to the ratio of the grid impedance to the inverter impedance. The grid impedance is dynamic, meaning that it differs when measured off-line (inverters not connected) and on-line (inverters connected). Furthermore, inverters introduce harmonics into the system as a result of the internal switching frequencies which can also affect the power quality and performance of the system[3].

In order to improve the stability, the system has to be identified in order to obtain suitable accurate wide-band (high frequency) models. This is done through Frequecy-Response Analysis (FRA), a type of system identification, which allows for wide-band signals to be injected into the system. System identification can be done through parametric methods or non-parametric methods depending on the objectives [4]. When the system does not require to be parametrized, non-parametric methods are more advantageous [5]. The models obtained from FRA can then be used for other applications such as power systems simulations, stability studies, spectral analysis fault detection and filter design. FRA has also been used for condition monitoring of power systems equipment such as transformers [6], [7], [8].

This project focuses on the design of an excitation signal suitable for power system identification. The signal has to have a wide frequency spectrum such that it can excite a power system over a wide frequency band and a response obtained. The response can then be used to analyse the behaviour of the system within the measured frequency spectrum. The signal is modelled from first principles (mathematical derivations) and then

implemented by building a circuit to generate the excitation signal.

## 1.2 Project Objectives

The aim of the project is to design and develop a low-voltage random impulse signal generating circuit. This circuit is used to generate a wide-band excitation signal suitable for power system identification. The objectives are as follows;

- To study system identification techniques, excitation signals, impulse generating circuits.
- To generate a Pseudo-Random Binary Sequence (PRBS) using a micro-processor.
- To control switches using a PRBS to generate a Pseudo-Random Impulse Sequence (PRIS).
- To perturb a power system using generated signal.
- To compare the characteristics of the developed practical impulse signal to theoretical analysis and simulations.

## 1.3 Project Outline

**Chapter 2** gives an overview of literature applicable to the project. It looks at the procedure of system identification, the types of signals used and the different impulse generating circuits.
**Chapter 3** provides a detailed procedure on the main circuit design based on its operation along with secondary circuits used in the project.
**Chapter 4** details the software approach of how the switching scheme of the main circuit is achieved.
**Chapter 5** presents the testing procedure and the findings of the simulations and measured data.
**Chapter 6** concludes by providing recommendations based on the results and discoveries made over the course of the project.

# Chapter 2

# Literature Study

## 2.1 System Identification

System identification is the development of a mathematical model of a dynamic system based on experimental data [9]. In the development of a mathematical model, minimal information is known about the internals of the system under observation. This is commonly referred to as the black-box approach due to internals of the system being masked from the observer. Figure 2.1 illustrates a black-box system, only the input and output are observable.


Figure 2.1: A black-box system

Based on these two parameters, a mathematical model of the system in the form of a transfer function can be built. On the opposite end of the system spectrum exists the white-box test, also known as the open-box where the internals of the system are known prior to testing. White box testing is predominantly used in software development rather than physical system identification. This is due to the approach used in validate a given system. The internal system parameters are already known beforehand, meaning that tests are used to confirm what is already known.

The mathematical models of black-box system are derived from experimental observation of predominantly Linear Time-Invariant (LTI) systems. Most practical systems are however, non-linear. Non-linearity implies that the system's input-output relation does obey the superposition theorem [10]. Compared to linear systems, the output of a non-linear system in response to a weighted sum of several signals is not the weighted sum of the responses to each of those signal [10]. Before a non-linear system can be modelled,

a linear system equivalent has to be considered [11]. The linear model is then assessed to see if it meets the end-user's requirements through non-linearity tests as found in [12]. A non-linear system is modelled as the sum of its linear and non-linear parts, this ensures that the non-linear model performs better than the linear model [11].

The mathematical models used for system identification are represented as Ordinary Differential Equation (ODE)s. This results in a finite dimension describing the system. A model can be developed in either the continuous time domain or the discrete time domain. A procedure is often followed in order to identify the system. It depends mainly on the objectives of the observer but most follow a generic procedure as follows [9];

1. Input/output selection
2. Collecting the data
3. Estimation of the parameters
4. Validation of the model

The procedure for system identification is not limited to these steps. It can be tailored to suit the system under observation. The following example illustrates how the procedure can be applied in a practical scenario. It involves investigating the load of an electric circuit which can be one or a combination of electrical components, namely the Resistor (R), Inductor (L) and Capacitor (C).

## Step 1

When selecting the input and output of the system, the base assumption is that they are unique and observable. In LTI systems, the input and output are usually represented in the form of a transfer function as given by equation 2.1

$$G(s) = \frac{Y(s)}{U(s)} = \frac{output(s)}{input(s)} \tag{2.1}$$

In order to determine the electric component in the black-box, a known signal is injected into the input and based on the output signal a conclusion can reached that allows a model to be built in order to verify it. For such a simple system, the choice of input can either be a voltage or a current, the same goes for the output. Given that the objective is to find the type of load, an input of a current signal and a voltage signal output are the signals of choice for observation. This leads to a direct approach for the model solution given that the transfer function will be an impedance. Figure 2.2 shows the setup for the experiment in the MATLAB environment.

The input signal is a sinusoidal current with frequency of 50Hz a peak of 1A. The output is measured across the load because it is a voltage.

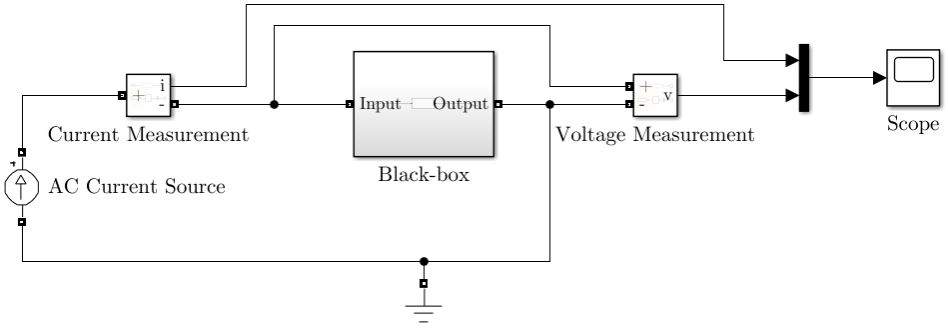



Figure 2.2: Black-box experimental set-up

## Step 2

The next step in the procedure is to collect the data. This is done by running the experiment, in this case applying the input signal to see how the output will be. The collected data is then presented in a format which can be interpreted by the observer as illustrated in figure 2.3

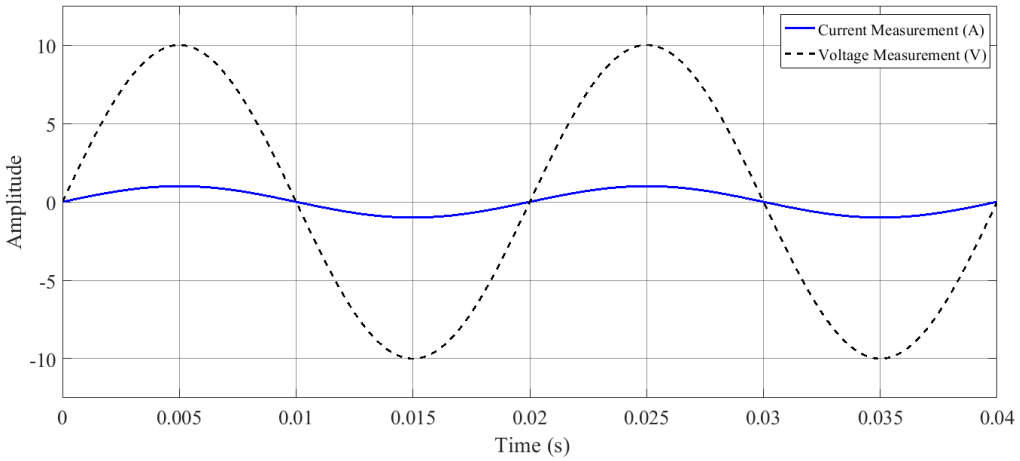



Figure 2.3: Black-box input and output

## Step 3

After collecting the data, the parameters of the model have to be estimated in order to build a model for testing. Three different loads are modelled, namely a resistor, an inductor in parallel to a resistor and a capacitor in parallel with a resistor. The values of these loads are determined by prior knowledge of such circuits. The initial values of the model can be refined depending on validation step outcome. The setup for the parameter estimation is shown in figure 2.4

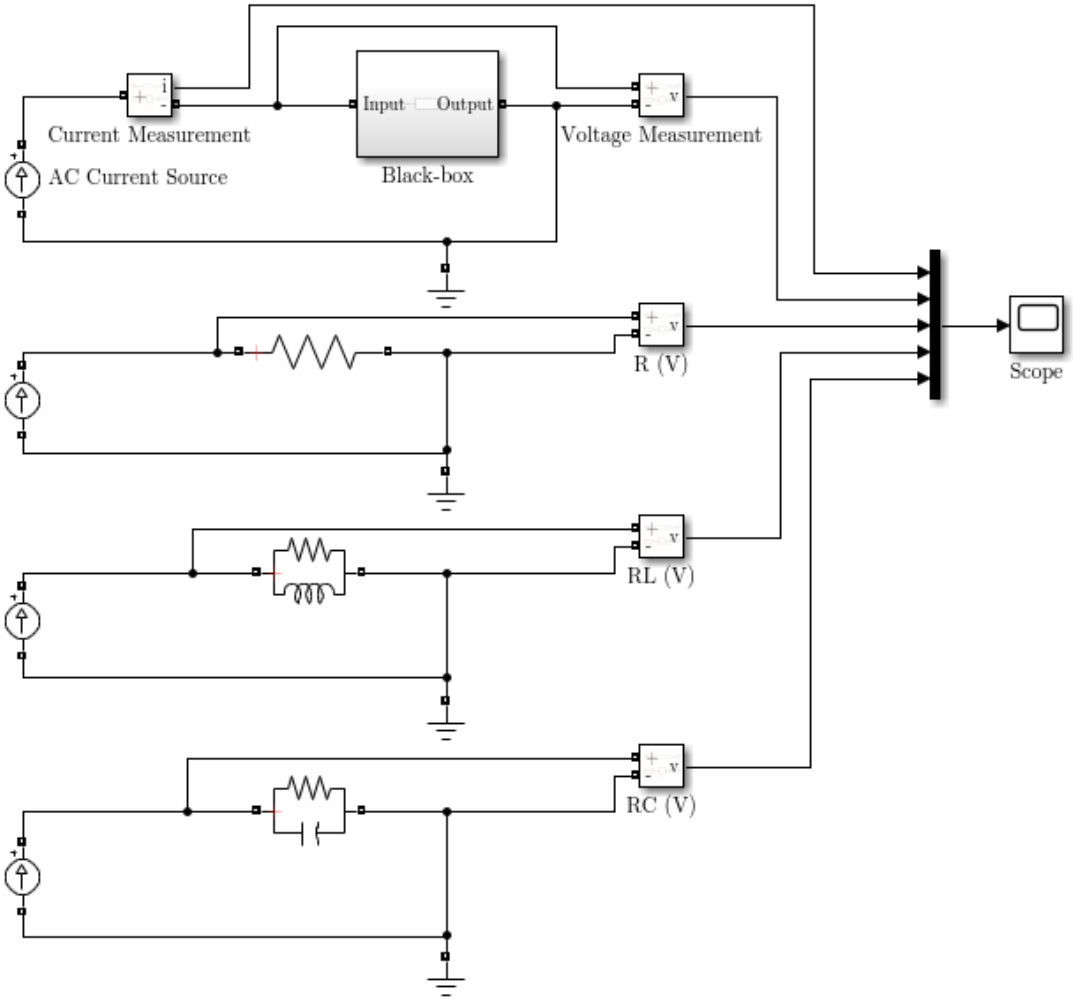



Figure 2.4: Black-box parameter estimation

## Step 4

The final step in the system identification procedure is to validate the model. This is done by repeating step 2 such that the outcome of the model can be compared to the previous data. This step is not an issue for LTI systems due to invariance but can be problematic in systems where the behaviour changes in a non-linear manner as a function of time. The outcome of validation step is illustrated in figure 2.5

Based on the output signals of the different loads, the purely resistive load closely resembles the black-box load albeit a different magnitude. The next step would be to refine the parameters again in step 3 such that they closely match the unknown black-box. Although this is a very simplistic approach into system identification, it gives an understanding of the procedure that is followed. The complexity of the procedure and experiments will depend on the system under investigation and the desired outcome.

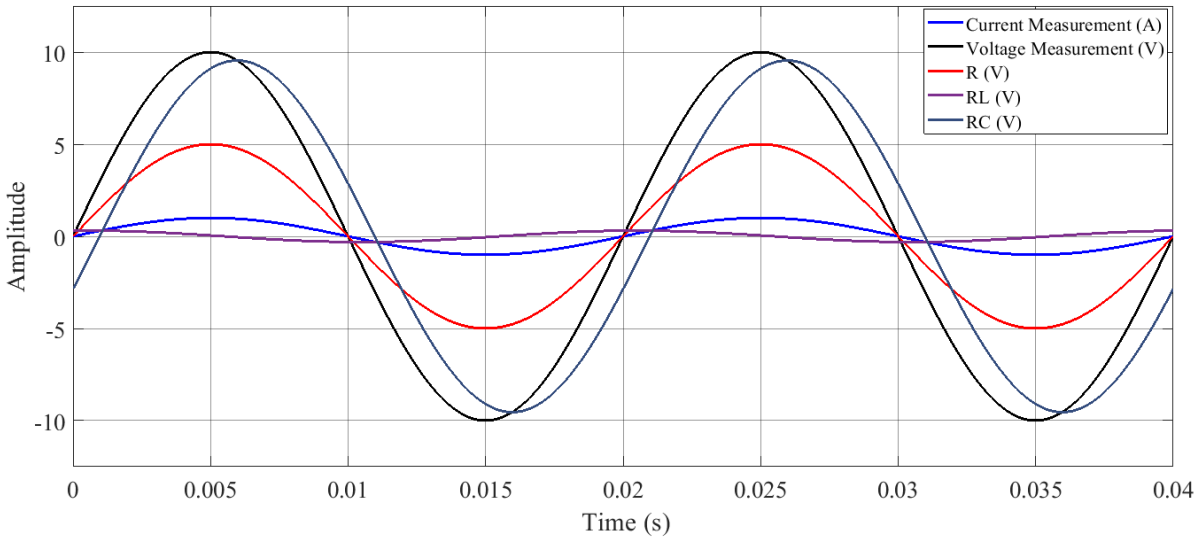



Figure 2.5: Black-box estimation output

## 2.2 Frequency Response Analysis

Frequecy-Response Analysis (FRA) is a tool that is used to analyse the results obtained from system identification in the frequency domain [13]. FRA can give a clearer overview of a system's response compared to a time-domain signal especially when there is a wide-band spectrum to be considered.

FRA is also used as a diagnostic technique mainly in the assessment of power transformer condition [8, 14]. Its primary use is to detect potential mechanical faults such as deformation in windings and core sheets [8]. FRA can be classified into the three stages as shown in Figure 2.6 [8]. The figure shows the connection of a transformer as a power system. The power system can degrade the components inside the transformer over time. Also worthy of note are uncontrolled external signals, mostly in the form of lightning which also affect the performance of the transformer.

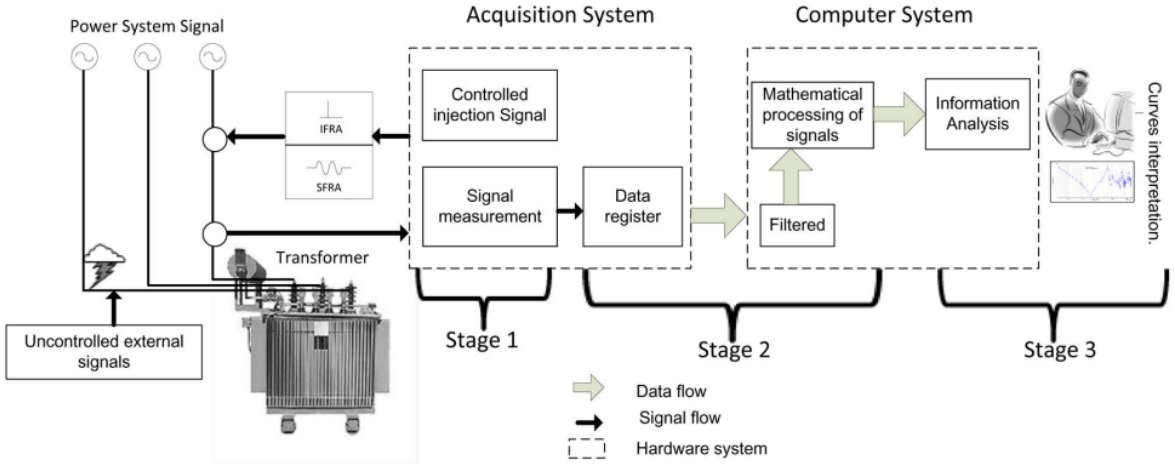



Figure 2.6: Scheme for the measurement of off-line/on-line FRA in transformers

The three stages are as follows;

1. Injection and excitation signal measurement.
2. Recording, filtering and processing of measured signals.
3. Curve analysis and interpretation.

In the first stage, the signal input can either be an Impulse Frequecy-Response Analysis (IFRA) or a Sweep Frequecy-Response Analysis (SFRA). In IFRA method, a single non-periodic signal is injected on the input side of the transformer and voltage is induced on the output terminals. A Fast Fourier Transform is carried out to obtain the frequency spectrum of both the input and the output. The ratio of the output to the input will give a transfer function. In SFRA, the initial step is similar to IFRA except that a Low-voltage (LV) amplitude, sinusoidal signal is applied across the input terminals. The signal is swept across a wideband frequency spectrum. The transfer function is then obtained the same way as before. DC signals cannot be injected into transformers due the following reasons;

1. DC signals are constant therefore they cannot induce an EMF on the secondary of the transformer $Vs = -Ns * \frac{d\phi}{dt}$. The direction of the current does not alternate meaning that the magnetic field lines are constant ($\phi$ flux is constant).
2. The magnetic circuit will get saturated and burn the windings as an inductor acts as a short circuit for DC signals.

FRA in transformer monitoring has two main drawbacks. Firstly, the transformer has to be disconnected for measurements to be taken. Before the transformer is shipped out, FRA is carried to get a baseline of the transformer characteristics meaning that when the test is carried out, it has to be disconnected to eliminate the influence of other systems connected downstream. As a result, condition monitoring is more of a corrective approach when a fault is suspected as opposed to a predictive approach which is more desirable. Secondly, it is currently difficult to obtain a clear diagnostic from the collected results due to lack of industry standards when testing. The physical errors that arise within the transformer do not have a distinct frequency footprints making it difficult to find the actual error.

Another application of FRA is in the identification of power system parameters such as network impedance or high frequency transformer parameters [15, 16, 17]. The procedure is similar to that of condition monitoring except that parameter estimations are carried to model the system under investigation as accurate as possible. The system to be model is represented as a two-port network as shown in Figure 2.7. The two-port network in most

power system identification applications involves finding the impedance, hence the use of current and voltage measurements. The current input to the system is measured across the frequency spectrum of interest. The voltage excitation is then measured, allowing for a transfer function to be calculated.

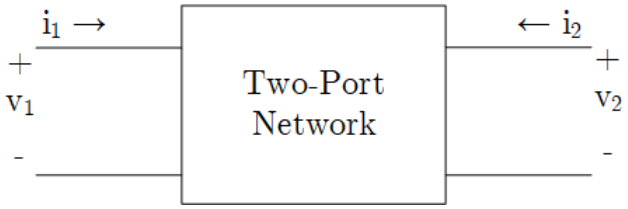



Figure 2.7: Two-port network of a system

## 2.3 Parameter Estimation

The impedance of a system tends to be function of frequency due to series inductance, shunt capacitance and resistive loads. Given that change of the overall impedance is non-linear, the parameters have to be refined in reference to the actual system behaviour. Figure 2.8 shows a feedback loop system that allows for parameters to be calculated. The model can be achieved using either state-space variables or transfer functions [15].

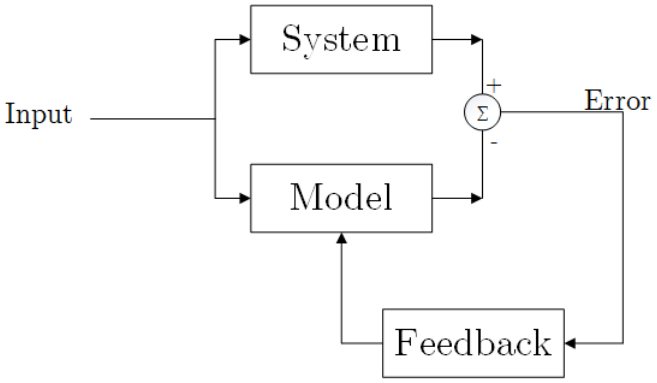



Figure 2.8: Generic feedback loop

A known input is applied to both the model and the system of interest. The difference between the outputs is the error by which the model needs to be corrected by. This value is fed into a controller which adjusts the states as appropriate and then feeds it back to model. This allows for the error in the parameter estimates to be lowered resulting in a more accurate model.

## 2.4 Excitation Signals

When analysing or identifying a system, a known signal is applied as an input in order to observe the response of the system. In some applications, the choice of the signal may be limited by environmental factors. For example, for serial communication, only binary data is used due to the system only recognising logic high or logic low. As such, applying a continuous signal of varying amplitude will not be suitable for such an application. Appropriate input signals have to selected in order to excite the system for sensible data to be collected. Excitation signals are classified into three different categories [18];

1. General Purpose (no optimization)
2. Test (fully automated optimization procedure)
3. Advanced Test

A wide range of signals can be generated with varying patterns and frequencies to identify the internals of a system [19]. The response of these signals is then referenced to prior knowledge gathered from past data in order to identify a given system. This study focuses on general purpose signals. There are two main quality measures of an excitation signal, the crest factor and the time factor.

### Crest Factor

The crest factor $Cr(t)$ is defined as the ratio of the peak value of the signal to its effective Root Mean Square (RMS) in the frequency band of interest [18].

$$Cr(t) = \frac{u_{peak}}{u_{rmse}} = \frac{\max|u(t)|}{u_{rmse}\sqrt{P_{int}/P_{tot}}}, \text{where } u_{rmse}^2 = \frac{1}{T}\int_0^T u^2(t)dt \tag{2.2}$$

The crest factor quantifies the compactness of the signal. Impulsive signals (large crest factor) will inject lower power than signals that have the same peak value and lower crest factor. The power in the frequency observation is expressed as effective RMS, $u_{rmse}$.

### Time Factor

The time factor $Tf(u)$ of a signal $u(t)$ is given by [18];

$$Tf(u) = \max \frac{0.5Cr^2(u)U_{rmse}^2}{|U(k)|^2} \tag{2.3}$$

It gives an indication of the power distribution of the signal over the frequencies of interest. For a signal to be considered optimum, it should have a low crest factor and time factor.

### 2.4.1 General Purpose Signals

General purpose signals refer to signals that do not have any form of specialized optimization. They are able to excite a system with a near flat power spectrum. Periodic excitation is mainly favoured due to little to no spectral leakage. Aperiodic signals tend to have leakage errors. Leakage occurs when the spectral resolution of the signal is low, as is the case for aperiodic signals. In periodic signals, the signal's length can be increased in order to reduce the impact of noise, a non-periodic signal. Noise will not appear in a periodic manner therefore, it can be extracted. In aperiodic signals, the data set is made larger such that the effects of noise in the system are minimized. As a result, to achieve a specific accuracy the measurement time has to increase. Time-limited signals (bursts) are exceptions to this rule due to the continuous spectra being sampled correctly with the Discrete Fourier Transform (DFT) provided the amplitude spectrum is sufficiently band-limited [18].

#### 2.4.1.1 Swept Sine

A swept sine (periodic chirp) is a sine sweep where the frequency is swept up and/or down in one measurement period, and this is repeated such that a periodic signal is created [20].

$$u(t) = Asin((at + b)t)\ 0 \leq t < T_0 \tag{2.4}$$

where, $T_0$ is the period, $a = \pi(k_2 - k_1)f_0^2$, $b = 2\pi k_1 f_0$, $f_0 = \frac{1}{T_0}, k_2 > k_1 \in \mathbb{N}$, and $k_1 f_0$, $k_2 f_0$ the lowest and the highest frequency, respectively. Figure 2.9 shows a time domain visualisation of a swept sine signal generated in MATLAB.

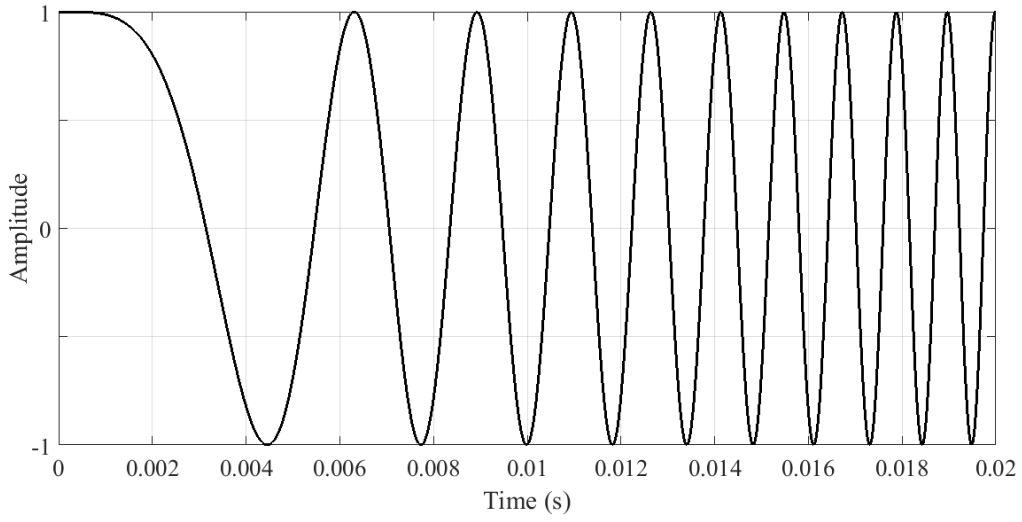



Figure 2.9: A swept sine signal

**Properties**

- Periodic signal with period $T_0 = \frac{1}{f_0} \rightarrow$ no leakage.
- Frequency resolution $\frac{1}{T_0}$.
- Most of the power is equally distributed in the user-selected frequency band $[k_1, k_2]f_0$ with $k_2 > k_1 \in \mathbb{N}$.
- Crest factor typically 1.45, time factor typically between 1.5 and 4.

Sine sweeps are characterised by a low crest factor but the magnitude is not flat over a wideband spectrum. As a result, frequency components with lower Signal-to-Noise Ratio (SNR) are introduced leading to longer time measurements. It is not possible to generate a signal with an arbitrary amplitude spectrum even though a band spectrum can be created. Another drawback is not only are the frequency lines of interest excited, but also a number of other spectral lines appear. This is not an issue with linear systems, but it can be very disturbing in systems with non-linear behaviour [18]. Sine sweeps yield high SNR signals which are desired for grid impedance estimation. The measurement renders them undesirable for on-line applications.

#### 2.4.1.2 Pseudo-Random Binary Sequence

A Pseudo-Random Binary Sequence (PRBS) is a random signal that takes on either a high state or a low state at discrete multiples of the clock period $T_c$ [21]. While it appears random, the states can be determined when the sequence generation scheme and initial conditions are known [22]. The autocorrelation of any given PRBS also reveals that it is periodic.

The PRBS$k$ or PRBS-$k$ notation (such as PRBS7 or PRBS-7) gives an indication of the size of the sequence [23]. In turn, the $k$ also shows the number of shift registers required. $N = 2^k - 1$ is the maximum number of bits in the sequence that will form a unique pattern before repetitions occur. $k$ also indicates the longest run possible of the same bit state. Table 2.1 shows the properties of a PRBS4 and PRBS14 sequence assuming the taps are configured for MLBS.

| Number of shift registers | k | 4 | 14 |
|---|---|---|---|
| PRBS Length | $2^k - 1$ | 15 | 16383 |
| Number of 1's | $2^{(k-1)}$ | 8 | 8192 |
| Number of 0's | $2^{(k-1)} - 1$ | 7 | 8191 |
| Number of edges | $2^{(k-1)}$ | 8 | 8192 |

Table 2.1: Properties of a PRBS4 and a PRBS14

**Properties**

- Periodic signal with period $T = NT_c \rightarrow$ no leakage.
- Frequency resolution $= \frac{1}{T}$
- Most of the power below $0.4f_c = \frac{0.4}{T_c}$ . Optimal choice of the clock frequency $fc = 2.5f_{max}$ , with $f_{max}$ the maximum frequency of interest
- Crest factor is 1 if all power is considered, time factor typically 1.5

A PRBS is based on a sequence of length $N$. Maximum-Length Binary Sequence (MLBS) is the most common of the sequences [18], [24], [25] used due to the simple implementation using Linear Feedback Shift Register (LFSR). A LFSR is a shift register that uses the previous state bit as the input. It is widely implemented using Exclusive Or (XOR) gates. Table 2.2 shows the truth table of an XOR gate.

| Input | | Output |
|---|---|---|
| A | B | |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Table 2.2: XOR gate truth-table

Figure 2.10 shows a LFSR of PRBS4 with the XOR connected to taps 3 and 4 of the shift register. The taps to which the XOR gates differ depending on the length of the desired PRBS. It can be a single gate or multiple gates depending on the taps in the polynomial. The seed (initial start) of the PRBS has to be non-zero. A seed of zero (all bits set to 0) will result in the 0s being fed back into the register rendering a non-functional LFSR.

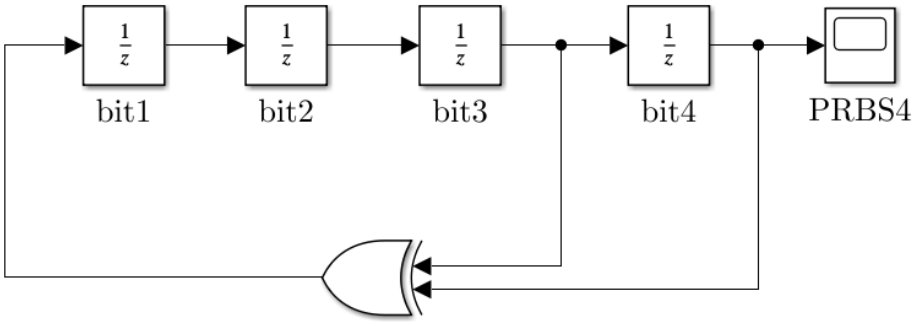



Figure 2.10: A LFSR of a PRBS4

As a result, the PRBS can be implemented using flip-flops and XOR gates in hardware. However such a set-up would limit flexibility in terms of the PRBS length because of the taps not being interchangeable across different lengths. Another issue with hardware implementation is clock frequency limitation as a result of an external crystal being used. This would also have to be fixed making it difficult to analyse wideband spectrum. Overall, hardware implementation poses numerous obstacles, especially debugging due to the high number of physical components and required PRBS logic. The alternative is to implement using software which gives more control for applications in terms of frequency and length.

Figure 2.11 [23] shows the waveform characteristics of PRBS4 signal. The seed of the LFSR is all 1s. The order in which the output comes out as, is primarily determined by the seed. As such all combinations of the PRBS will be realised over a MLBS period regardless of the seed, provided it is non-zero and that the taps are set for MLBS.

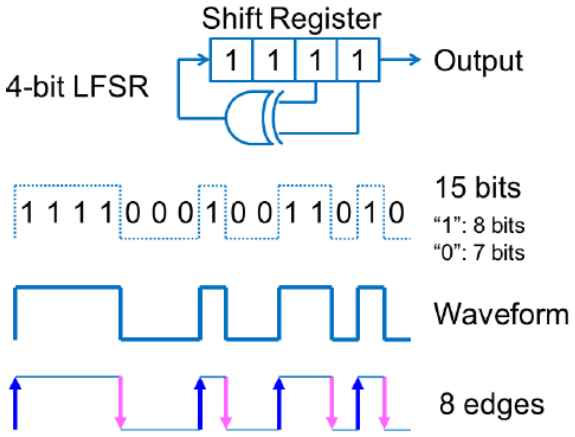



Figure 2.11: PRBS4 waveform

#### 2.4.1.3 Impulse

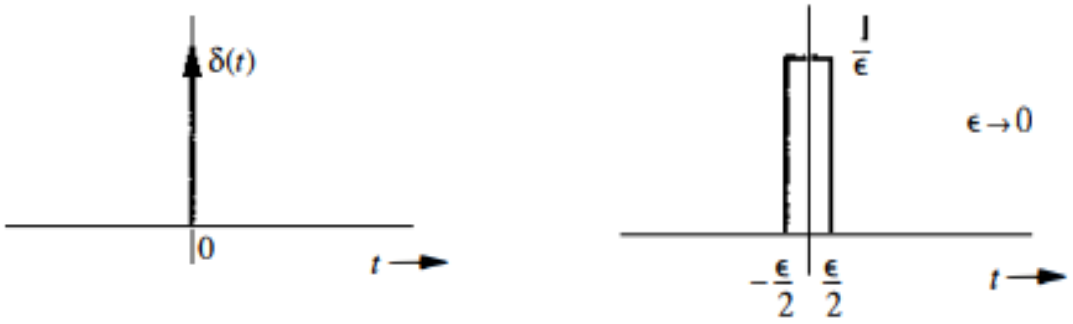
Another useful signal in system identification is the unit impulse signal also known as the *Dirac Delta*. Mathematically, the unit impulse is defined as follows;

$$\delta(t) = 0,\ t \neq 0 \tag{2.5}$$

$$\int_{\infty}^{\infty} \delta(t)dt = 1 \tag{2.6}$$

The impulse can be viewed as a tall narrow pulse of unit area as shown in Figure 2.12 [26]. The width of the rectangular pulse is a small value, $\epsilon$, the height is the reciprocal of the width, $1/\epsilon$ as $\lim_{\epsilon \to 0}$ as shown in Figure 2.12b. The unit impulse can therefore be regarded as a rectangular pulse whose width becomes infinitesimally small as the height increases. Figure 2.12b shows the simplified version of an impulse. The arrow indicates the high amplitude which is the inverse of the signal. The overall area of the impulse

remains constant at unity [26]. A pure impulse signal that matches the mathematical equivalent cannot be realised in a physical sense.




(a) Unit Impulse (b) Unit Impulse Approximation

Figure 2.12: Unit impulse signal and its approximation

In practical high voltage applications, impulse signals are used to simulate lightning in the testing of equipment such as power transformers, insulators and power cables [27]. The features of the impulse are fast rise times relative to the slow fall time. A practical impulse is shown in Figure 2.13 [28] with rise time given as $t_{rise}$ and the fall time as $t_{tail}$.

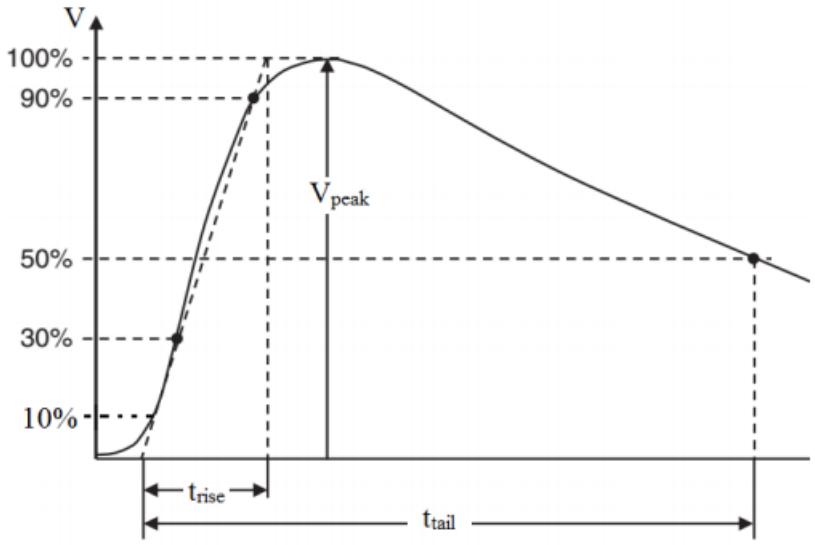



Figure 2.13: Practical Impulse Signal

When used for testing high voltage equipment, the impulse signal has to adhere to specific time and amplitude criterion as defined in IEEE Standard 4-2013 [29]. Although the signal waveform is specific to voltages, it can also be applied to current impulses. The practical impulse can be described mathematically by the following equation;

$$V(t) = V_0 K(e^{-\alpha t} - e^{-\beta t}) \text{ [29]} \tag{2.7}$$

where $V_0$ is the initial amplitude, $K$ is th amplitude modifying factor, $\alpha$ and $\beta$ are the time constant of the impulse. The time constants determine how damped the signal will

be, therefore determining the response and oscillations within the system. The damping ratio, $\zeta$, describes how the oscillations will decay as a function of time. The damping ratio focuses primarily on second-order systems such as an impulse signal or RLC circuits. The damping ratio can be classified into three categories, **underdamped** ($\zeta < 1$), **overdamped** ($\zeta > 1$), **critically damped** ($\zeta = 1$). The effect of the damping ratio for a second-order system is illustrated in Figure 2.14 [30].

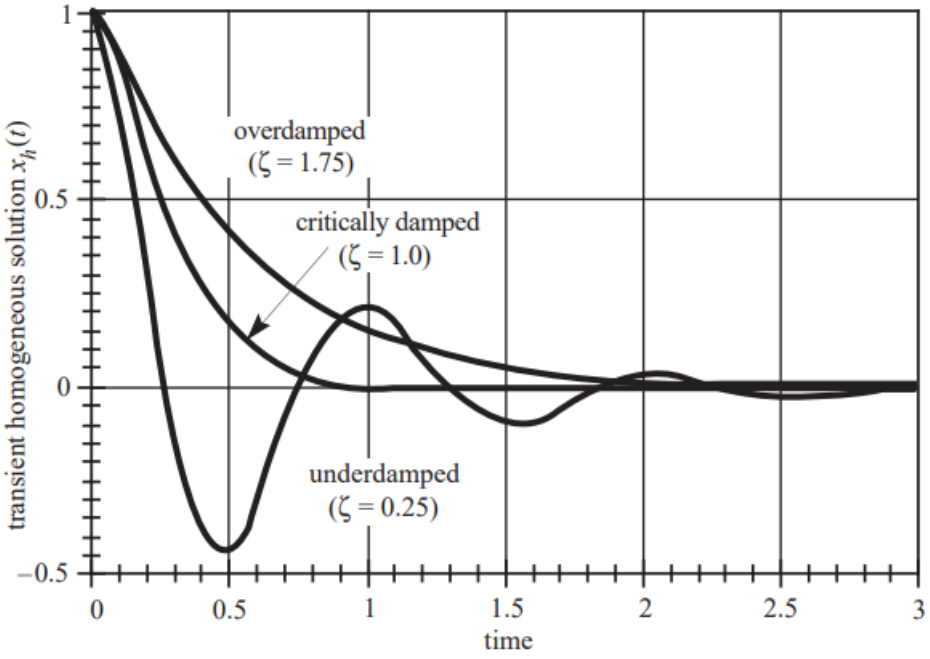



Figure 2.14: Effect of damping on system response

It can be seen that the critically damped system reaches a steady state quicker compared to the overdamped and underdamped systems. It also does so with no oscillations. This is a sought after signal property when testing power systems as it will not introduce extra undesired oscillations. In most cases, the damping ratio solution is purely mathematical and can not be realised with physical components. As a result most designs tend to be slightly over as this will not lead to oscillations within the system.

## 2.5 Summary

In this project, the signal required is a bipolar current Pseudo-Random Impulse Sequence (PRIS). It is a PRBS in the form of current impulses. The pseudo-random nature of the impulses allows for different frequency sequences to be injected to the system. The randomness of sequences prevents a build up of harmonics on the measured signals which will lead to better identification. The average current that is injected into a system tends to zero due to the current value varying between positive and negative cycles as a result

of the PRBS properties. As a result of the impulse switching, high frequency components, similar to those found in control systems for renewable energy sources, such as inverters, are injected into the system in a controlled manner.

# Chapter 3

# Hardware Design

## 3.1 Overview

The hardware aspect of this design centres around the current excitation generating circuit with supporting independent power supplies. The components used in the design should adhere to the specifications listed in Table 3.1.

| Specification | Value |
|---|---|
| Voltage rating | 1kV |
| Current rating | 10A |
| Frequency rating | 100kHz |

Table 3.1: Design Specifications

Figure 3.1 shows the block diagram of the whole system. The PRBS signal is generated in software using a micro-controller and it is discussed further in Chapter 4. The rest of the system diagram is hardware-based and it is discussed further below.

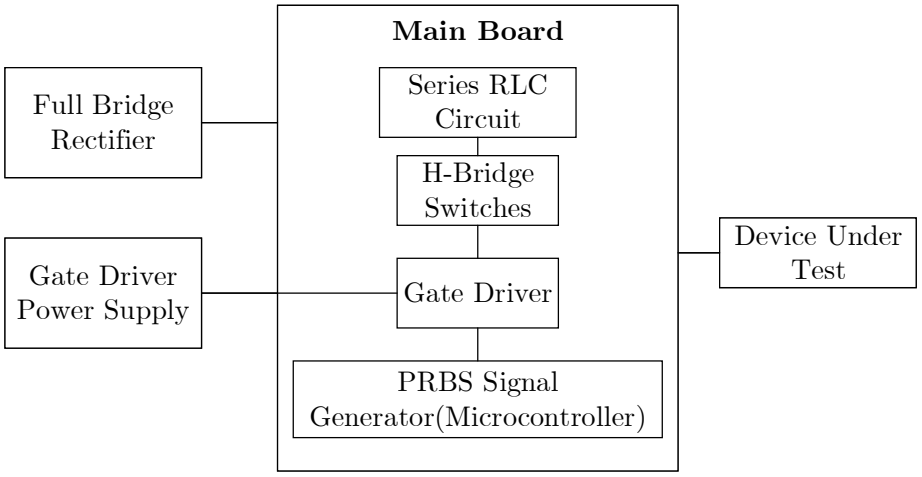



Figure 3.1: Block Diagram

## 3.2 Operation of circuit

The operation of the circuit involves charging a capacitor up in one direction and then releasing impulse currents when the polarity of the voltage across capacitor is changed. A layout of the high level circuit is shown in Figure 3.2. The use of a DC source means that R and L only drive current in one direction, even as the switch occurs.

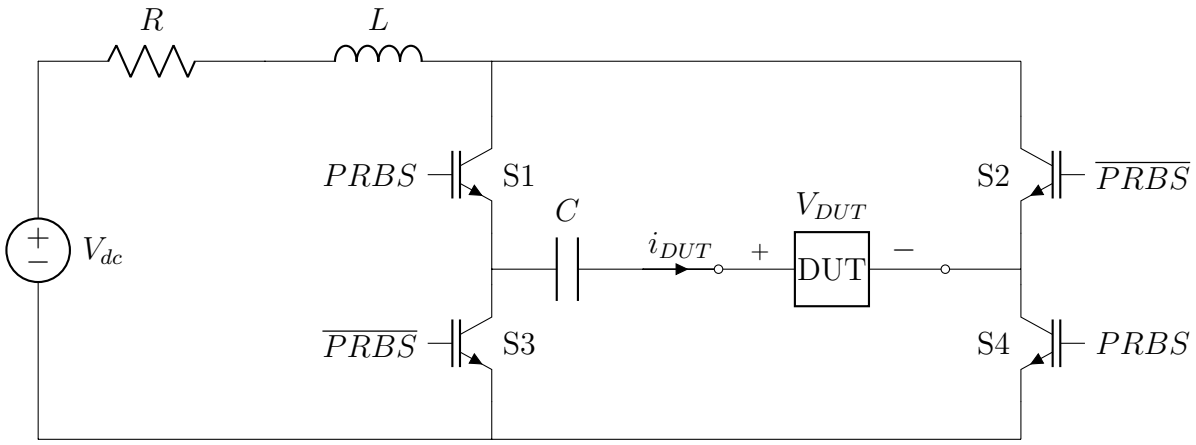



Figure 3.2: Excitation Current Generating Circuit

The switches in the circuit are laid in an H-bridge configuration with each half bridge pair switching complimentary (S1 and S4 are both ON when S2 and S3 are OFF). The Device Under Test (DUT) is placed in series with the capacitor such that a closed loop can be formed when the switches are operational. In the first instance when S1 and S4 are both ON, the capacitor will charge up due to the difference in voltage potential on its terminals. When S2 and S3 turn on, the voltage across the capacitor does not change instantaneously, rather its reference frame as seen by the DUT changes due one of capacitor terminals being connected to ground. As a result, current impulses will develop as the capacitor discharges into the device. The H-bridge configuration is what results in bipolar current impulses as the circuit continuously switches.

### 3.2.1 Impulse Generator Circuit

The aim of the project is to design a signal suitable for power system identification through excitation. The high frequency components generated by control power electronics inject current as opposed to voltage into the system. This is supported by Kirchhoff's circuit laws.

From Kirchhoff's Current Law (KCL), it can be seen that it is simpler to excite a system using current. When wideband currents are injected, voltages will be induced in the system that correspond to the excitation current, subject to the impedance of the system. When generating impulse currents, capacitors tend to be used as they can

store charge. As they discharge, impulse currents as released instantaneously. The most common circuit for generating impulse currents is a series RLC circuit as shown in Figure 3.3.

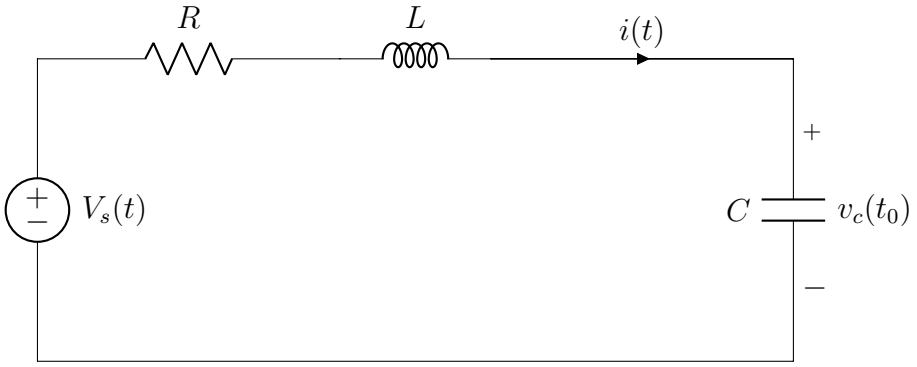



Figure 3.3: Series RLC circuit

The solution of the series RLC circuit is fully derived in Appendix D and the solutions are summarized in Table 3.2.

| Damping | Roots | Current Solution, $i(t)$ |
|---|---|---|
| Overdamped | $\omega_0^2 < \alpha^2$ | $A_1 e^{s_1 t} + A_2 e^{s_2 t}$ |
| Underdamped | $\omega_0^2 > \alpha^2$ | $B_1 e^{-\alpha t} cos(\omega_d t) + B_2 e^{-\alpha t} sin(\omega_d t)$ |
| Critically damped | $\omega_0^2 = \alpha^2$ | $D_1 t e^{-\alpha t} + D_2 e^{-\alpha t}$ |

Table 3.2: Series RLC current solutions

As discussed in Chapter 2.4.1.3, a critically damped response is desired in most system however this is near impossible to realise in practice. The solution is to design using a slightly overdamped response. The overdamped current solution is used to size the RLC components.

#### 3.2.1.1 Impulse Waveform

The impulse waveform of design is a modified form of the standard impulse waveform (Figure 2.13). In Figure 3.4, the $T_R$ and $T_f$ are used to represent the rise time and fall time respectively. The numerical values are derived from the exponents of the overdamped response current solution.

A PRBS14 at 15kHz is used as the switching signal in order to determine numerical values for the $T_R$ and $T_f$. To inject high frequency current components into a system, a small rise time is required due to the inverse proportionality of time and frequency ($f = 1/T$). $T_f$ should be long enough such that the impulse does not reach steady state

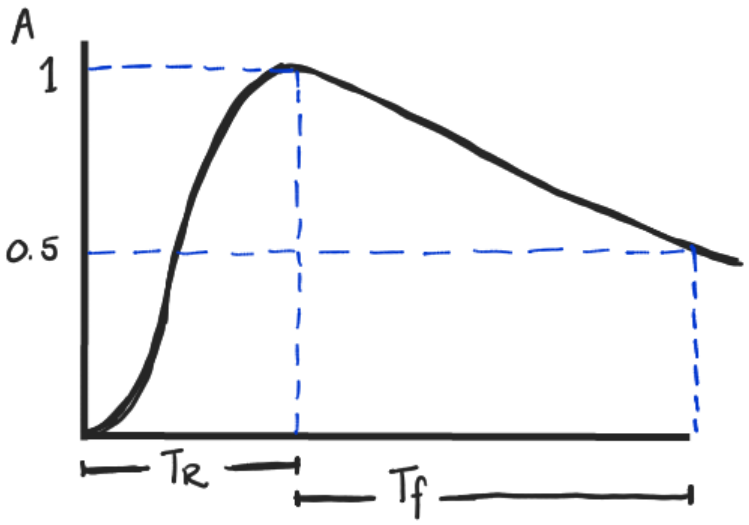



Figure 3.4: Impulse Waveform

on the longest high-state sequence run of the switching signal. On short sequence runs, the current signal will be cut off, further increasing the wideband spectrum of the signal.

Based on the rise time and fall time specifications, an equation based on the time constants is formulated as shown in Equation 3.1. The $5\tau$ principle is used as this signifies a signal reaching 1% of its original value. For the rise time, $5\tau_R$ means that the signal is at peak steady state, while for the fall time, $5\tau_f$ is the time it takes to decay. The summation is made equal to the longest PRBS14 sequence run. It can be seen that $5\tau_R$ will be greater than $5\tau_f$ when the rise time is smaller than the fall time.

$$5\tau_R + 5\tau_f = 14 \times T_{clock} \tag{3.1}$$

#### 3.2.1.2 RLC Component Sizing

The constants $\tau_R$ and $\tau_f$ are functions of R,L and C therefore one of the values have to set in order to balance the unknowns and the number equations. In order to streamline the calculations, a script was used to determine the size of the components. The script used can be found in Appendix I. The capacitor was chosen as $2\mu F$. The obtained RLC values are summarized in Table 3.3.

| Component | Manufacturer | Script Value | Practical Value |
|---|---|---|---|
| Resistor | LTO150F100R0JTE3 | 93.33 Ω | 100 Ω |
| Inductor | RS Pro 1048448 | 62 $\mu F$ | 100 $\mu F$ |
| Capacitor | C4GAMUD4200AA1J | 2 $\mu F$ | 2 $\mu F$ |
| | $\tau_R$ | $-1.8600 \times 10^{-4}$ | $-1.9899 \times 10^{-4}$ |
| | $\tau_f$ | $-6.6667 \times 10^{-7}$ | $-1.0051 \times 10^{-6}$ |

Table 3.3: Series RLC Values

Compared to the script values, R was increased in order to overdamp the response such that the possibility of oscillations was minimized. L was also increased to minimize the impact of increasing the damping ratio by R as this can result in a slow response of the system. There is a trade off between signal amplitude and damping in a system. Excessive damping can lead to lower signal amplitudes resulting in a low SNR leading to errors in measurement. The damping ratio is given by Equation 3.2 and the circuit is overdamped with a ratio of $\zeta \approx 7.071$.

$$\zeta = \frac{R}{2}\sqrt{\frac{C}{L}} \tag{3.2}$$

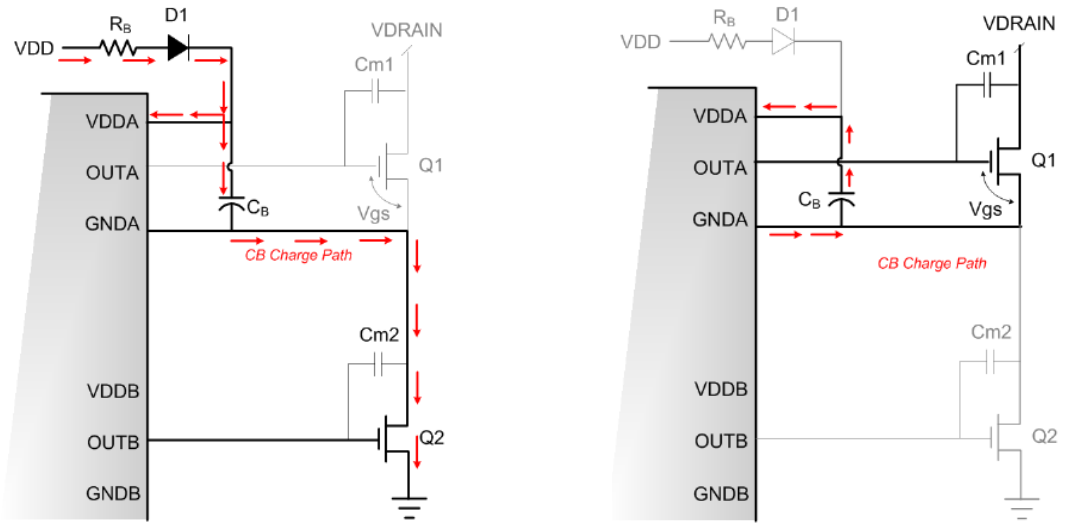
#### 3.2.1.3 Switches

The IXGH24N170 Insulated-Gate Bipolar Transistor (IGBT) [31] is used at the switch in the circuit design because it met the design specifications while being the most cost effective, locally available switch.

### 3.2.2 Bootstrap Design

IGBTs are used for switching the circuit. This is due to their ability to pass higher currents in their ON-phase compared to BJTs, also while switching at high frequencies. IGBT are a combination of a MOSFET and a BJT. The switch is turned on by applying a voltage to the gate relative to the emitter. The supply source of the gate voltage is independent of the voltage bus that appears on the collector and emitter pins. A gate driver is used due to the relatively low voltages that come from the logic of the micro-controller (5V) where as the required gate voltage is about 15V. When the switch is turned off, a large voltage can be blocked depending on the voltage at the collector node [32]. If the voltage applied to the collector is greater than the gate driver supply, the charge required at the gate will increase given that the emitter does not sit at ground for the high side switch of the bridge. A bootstrap circuit has to be used such that the operating point of the gate is raised allowing the gate drive supply to remain functional. A bootstrap circuit consists of three components, a capacitor, resistor, and a diode. Figure 3.5 [33] shows the functionality of a bootstrap in charging and sourcing mode.

When the low side switch, $Q2$, is on and $Q1$ off, $C_B$ is charged by the current flowing from the supply $VDD$ to the ground node via $R_B$, $D_1$ and $Q_2$ as shown in Figure 3.5a. When $Q_2$ switches off and $Q_1$ switches on, the voltage at the source of $Q_1$ will rise rapidly to its drain voltage. Figure 3.5b shows the circuit equivalent when $C_B$ is sourcing the current. At this point the source voltage at $Q_1$ will be greater than $VDD$ and will be blocked by the $D1$. $C_B$ is referenced to the source of $Q_1$ meaning that a bias voltage of $VDD$-0.7V will be maintained between the source and gate of $Q_1$ via $OUTA$ pin.




(a) $C_B$ Charging (b) $C_B$ Sourcing

Figure 3.5: Bootstrap Circuit

$R_B$ is used to lower the peak current in the system, while $D_B$ blocks the voltage in the reverse direction when the $C_B$ is supplying the charge. The values for the bootstrap components were determined using the application notes [33] the full calculations are given in Appendix D. Table 3.4 gives a list of the calculated values.

| Component | Value |
|---|---|
| Capacitor | $3.3\mu F$ |
| Resistor | $2.2\Omega$ |
| Diode | UF1007 |

Table 3.4: Bootstrap Components

## 3.3 Secondary Circuits

The system design also features two independent power supplies that provide different power for the gate driver an the main board.

### 3.3.1 Full Bridge Rectifier

The magnitude of the current impulses generated by the RLC circuit depend primarily on the magnitude of input voltage. DC voltage is used as the input to the current excitation generating circuit. A standard 30V bench supply could be utilised to power the main board, however the bootstrap circuit raises the node voltage between the emitter of high side switch and collector of the low side switch. The node voltages sits at about the required gate drive voltage (20V), this is also true for the second half bridge circuit. As such, the top voltage bus sits at about 40V meaning that a high input is required for the

current to flow in the desired direction through R and L. To produce the required DC voltage, a full bridge rectifier is used in this design as shown in Figure 3.6.

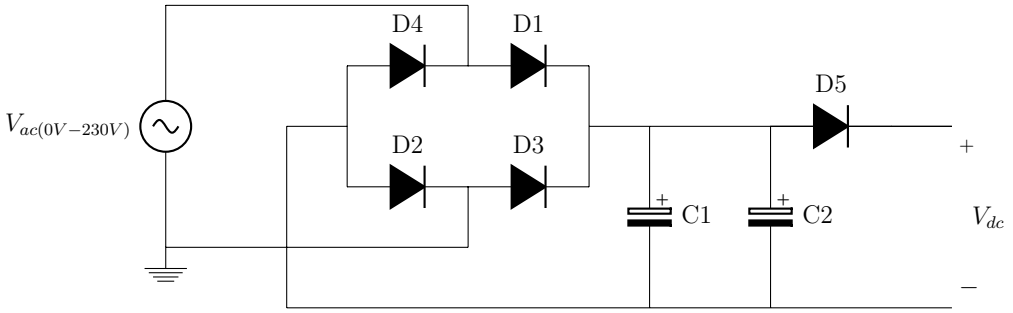



Figure 3.6: Full Bridge Rectifier

A blocking diode $D5$ is included in the design to prevent current flowing in the reverse direction from the excitation generating circuit as this can charge the capacitors up. The capacitors have to withstand rectified mains voltage ($325V_{dc}$) while minimizing the voltage ripple. There is a trade-off between filtering and capacitance at high voltages. As the voltage rating increases, the capacitance decreases for commercially available capacitors that can be used PCB designs. The diodes also have to withstand high reverse voltages such that they don't enter into their breakdown region while reverse biased. The full bridge rectifier provides physical isolation of the output from the input. The isolation prevents the AC source from being perturbed while a device is under test. This in turn will reduce distortion on the DC voltage bus.

### 3.3.2 Gate Driver Supply

The use of isolated gate drivers increases the number of power supplies required in the design. Gate drivers need two supplies, one for the input and another for the output. Both of these supplies have to be isolated from each other in order to maintain the isolation established internally in the chip. Figure 3.7 [34] shows the layout of the chosen isolated driver, the SI8233BB-C-IS1.

The input side ($VDDI_1$ and $VDDI_2$) requires a low voltage as it just handles logic signals from a micro-controller. The $5V$ on the Nano is used to power the input side of both gate drivers as they do not draw high current.

The output side of a gate driver ($VDDA$ and $VDDB$) require a high voltage relative to the input due to the voltage required to turn the switch on at the gate terminal. The turn on voltage differs for most IGBTs meaning that the output of the gate driver has to be variable within the turn on range. From the datasheet of the gate driver [34], the voltage required on the output side is between $6.5V$ and $24V$ while the test conditions of the IGBT are at $15V$ [31]. The required supply voltage has to also be in line with the

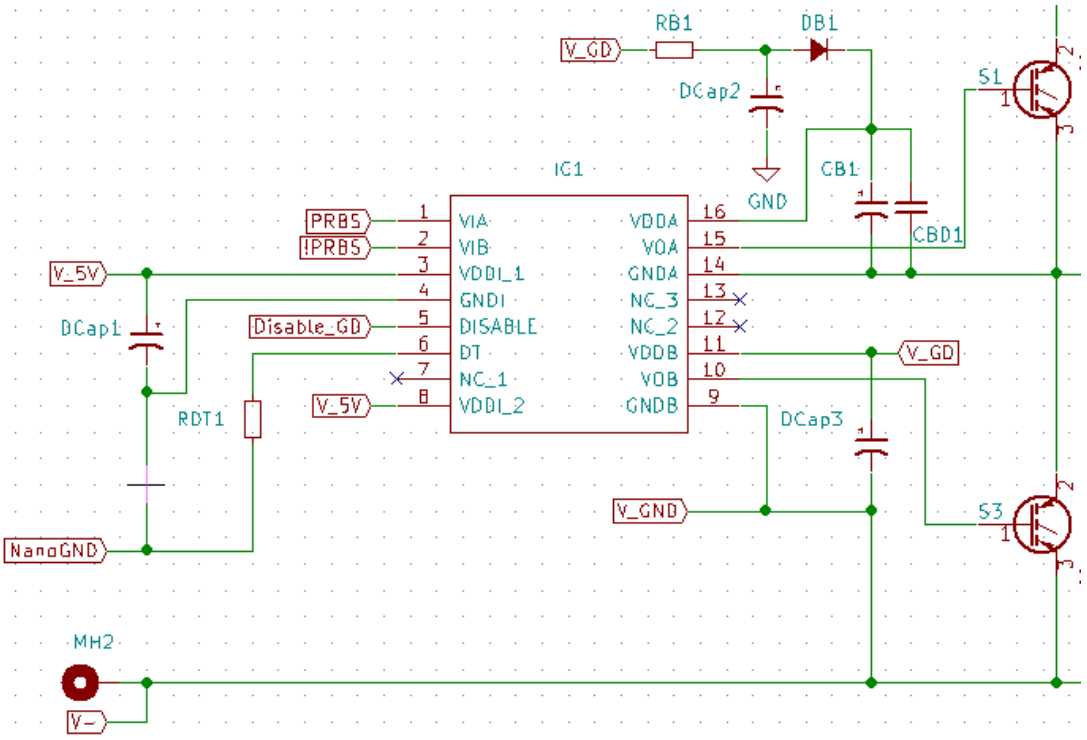



Figure 3.7: Isolated gate driver schematic

bootstrap design in Appendix D.2. The SR086 voltage regulator is chosen as the supply source for the gate driver output side due it being an adjustable switching regulator within the required voltage range [35]. Figure 3.8 shows the suggested circuit layout of the regulator.

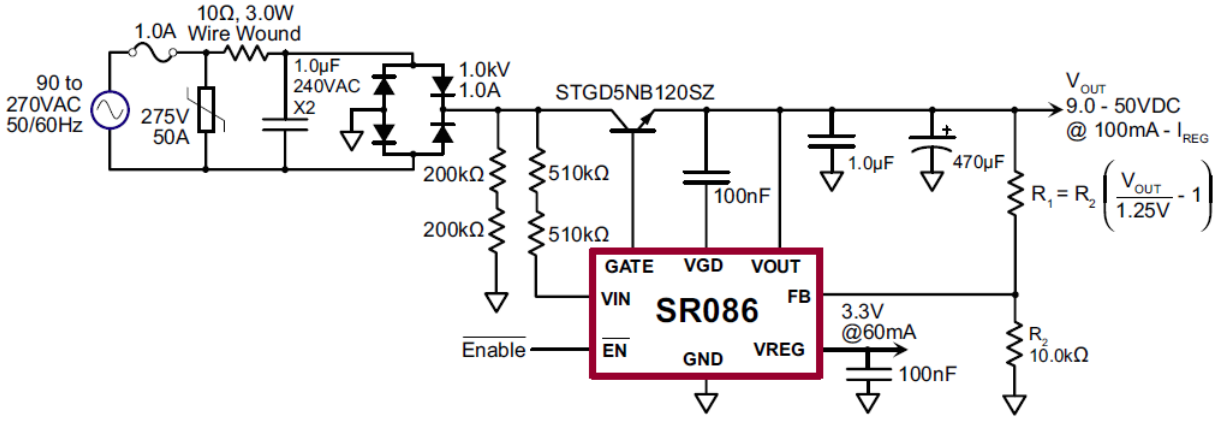



Figure 3.8: Adjustable voltage regulator

## 3.4 Other components selection

Binding posts with spade terminals were chosen as the anchoring points for the power supply and load. The spade connection has more contact surface area than a banana terminal. The current injected into the load consists of high frequency components meaning

that there is a high possibility of the skin effect occurring. The spade connections provide enough surface area for current to flow at high frequencies.

## 3.5 PCB Design

Printed Circuit Board (PCB)s were designed based on the circuit schematics in Appendix E. The width of the power traces where based on the IPC2152 [36] standard which requires external traces that carry current of 10A to be 7.62mm thick. KiCAD was used as the development environment.

## 3.6 Physical Circuits

The built physical circuits are illustrated in Appendix E

# Chapter 4

# Software Development

## 4.1 Overview

This section covers the software used directly on the main circuit. The main function at the core of the system functionality is discussed according to its operation. Software functions for data manipulation are covered in Chapter 5.

## 4.2 Micro-controller

In order to develop the control algorithm of the switches, a micro-controller is required. The PRBS logic stems from flip-flops and XOR gates which are native commands in the assembly language. Timing functionality is another requirement for the design due to wideband switching spectrum being required to generate current impulses over a wideband. The interface has to be handled by a Graphical User Interface (GUI) over a serial serial communication. However, assembly has limitations when it comes to serial communication, therefore the controller of choice has to be able to handle assembly and serial communication simultaneously. An Arduino Nano was used as the development board due to its small form factor and its capability of handling serial communication and timing functionality.

## 4.3 Flowchart

Figure 4.1 shows the software operational flow of the system. The data is entered on the GUI and then transmitted to the Nano through serial communication. The serial communication steps can be found in the GUI code in Appendix H. The data is then processed on the Nano before the switching occurs. The core of the software is the bitShifter() which is responsible for generating the PRBS sequences.

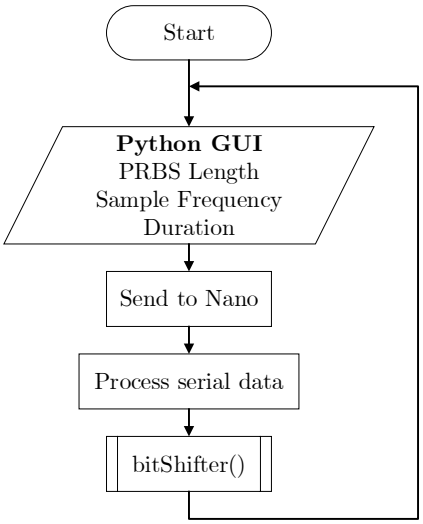



Figure 4.1: Software Flowchart

## 4.4 PRBS Signal

The PRBS is generated according to the logic found in Chapter 2.4.1.2. XOR gates are implemented using binary arithmetic inherent to the assembly language while the flip-flops are implemented using variables, which is similar to using a register. The function which does the shifting is shown in the code snippet in Listing 4.1.

```
1   lsb = lfsr & 1; /* Get LSB (i.e., the output bit). */
2   lfsr >>= 1; /* Shift entire PRBs sequence */
3
4   if (lsb == 1) { /* Only apply toggle mask if output bit is 1.
        */
5    lfsr ^= taps; /* Apply toggle mask, value has 1 at bits
        corresponding to taps, 0 elsewhere. */
6    // PRBS = 1, !PRBS =0
7    PORTC ^= B00010000;
8    PORTB  = B00000100;
9   }
10  else{
11   // PRBS = 0, !PRBS =1
12   PORTC ^= B00010000;
13   PORTB  = B00001000;
14  }
```

Listing 4.1: bitshifter()

In Line 1, the lsb (output bit), is the logical AND of the LFSR and a 1. This makes the output equal the desired output bit. In Line 2, the entire LSFR is shifted once to the

right with the result being assigned to itself, which is the equivalent shifting the flip-flops. The taps variable represents the locations of XOR gates in LFSR. Line 5 takes the XOR of the LFSR to the taps only if the output bit was a 1.

To control the switches, port manipulation was used on the Nano [37]. Port manipulation functions by accessing the port registers directly which allows for faster manipulation to the input and output pins. The $PRBS$ and $\overline{PRBS}$ signals are located in the PORTB register. When the lsb bit is 1, the pin corresponding to $PRBS$ is set high while $\overline{PRBS}$ is set low. The opposite occurs when lsb is 0. The $PRBS$ and $\overline{PRBS}$ should never be logic high simultaneously as this will lead to short circuiting on the physical board.

The XOR gates are located at different places for different PRBS length. As a result, the binary polynomials will differ in software [38]. The use of the bitshifter() function allows for expansion to include longer length PRBS due to the way it handles the data arbitrarily regardless of length. The polynomials are initialised as variables in the full code and can be found in Listing G.1.

The software handles PRBS signal generation from $PRBS4$ to $PRBS14$, a sampling frequency spectrum from 10kHz to 100kHz. The accuracy of the timing is limited by the pre-scaling of the Arduino. The run time duration is limited by the length of *long* type, which is 32 bits and the roll over of *Micros* function, which overflows every 70 minutes. The entire code is listed in Appendix G.

## 4.5 Python GUI

The GUI handles the graphical interface between the switches and the Nano. The commands are passed using serial communication which is available in Python and Arduino. Figure 4.2 shows the user interface used in the design.

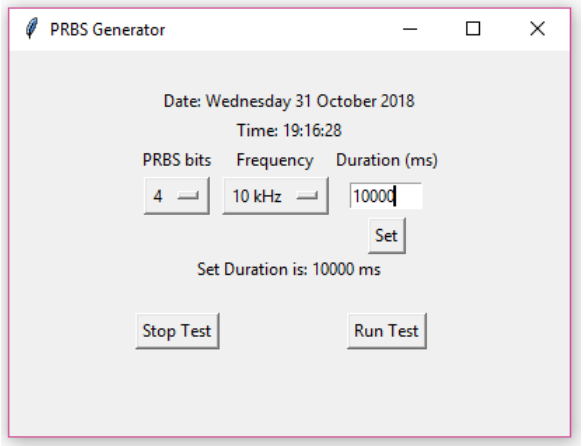



Figure 4.2: Python GUI

The GUI is kept as simple as possible in order to be user friendly. The PRBS bits and frequency fields correspond to the initialised lengths in the Arduino code. The duration field is a text box that allows any required time to be input in milliseconds. The error handling is done in Arduino due to the complexity of Python error checks. If an invalid time is passed, the serial buffer contents are emptied and the switches remain disabled. The GUI also has an option to interrupt a test should the length be deemed too long and for safety reasons. The GUI code can be found in Appendix H.

# Chapter 5

# Results and Measurements

This chapter documents the results of the simulated circuits along with their physical counterparts. Off-line system identification results are also measured in order to determine the accuracy of the excitation signal in identifying a given power system. The test conditions for all cases are tabulated in Table 5.1, unless stated otherwise.

| Specification | Value |
|---|---|
| Sample Frequency | 15 kHz |
| Excitation Signal | PRBS14 |
| Input Voltage | 60V |

Table 5.1: Test Conditions

Table 5.2 contains the equipment used to test the main circuit. The equipment affects the quality of obtained results in terms of doing PSD analysis. The limiting device is the current probe due to its bandwidth of 100kHz. The impact is that at frequencies close to 100kHz, the obtained will contain errors.

| Device | Model |
|---|---|
| Oscilloscope | Tektronix DPO2004B |
| Current Probe | Tektronix A622 |
| Differential Probe | GW Instek GDP-025 |

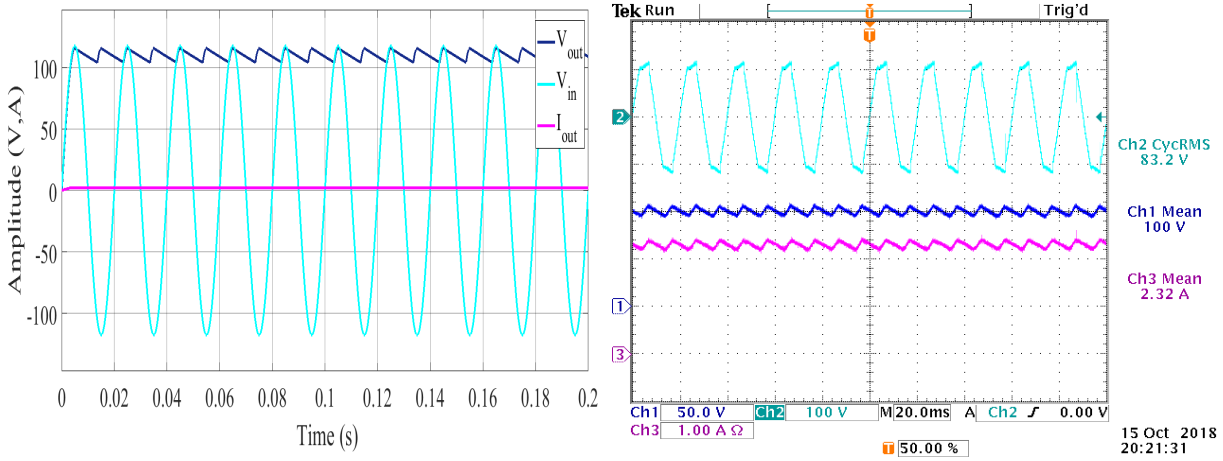
Table 5.2: Test Equipment

## 5.1 Isolated Power Supplies

The power supplies are physically isolated from earth ground in order to prevent the main circuit from being part of the system, should on-line tests be carried out. The isolation also stops the input and outputs of the power supplies from sharing the same ground which can lead to short circuits, especially on the diode rectifiers.

### 5.1.1 Full Bridge Rectifier

The function of the full bridge rectifier is provide a stable DC voltage output at high currents. To test this, the input was increased until a mean voltage output of $100V_{dc}$ was obtained. Figure 5.1 shows the comparison between the MATLAB simulation and the measured output. At $100V_{dc}$ output, the current through a test load of $50\Omega$ was about 2.3A with significant ripple as shown in Figure 5.1b. The ripple was due to the size of the output filter capacitors being relatively small. The size was limited by commercial availability. This simulation output in Figure 5.1a shows a high output for the same input due to line losses and diode losses not being simulated.




(a) Simulation (b) Measured

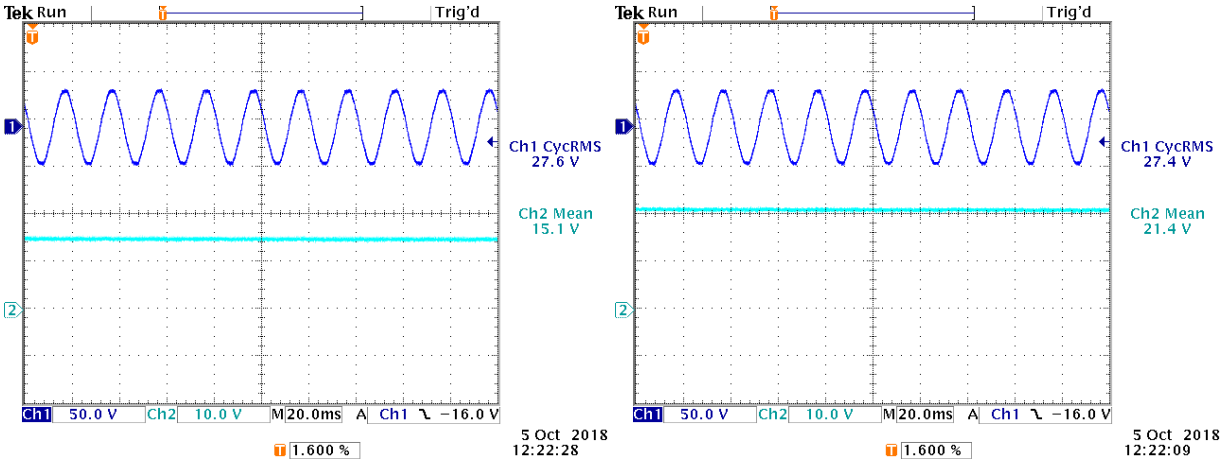
Figure 5.1: Full Bridge Rectifier

### 5.1.2 Gate Driver Supply

The designed gate driver supply had an output DC range between $15V$ and $22V$. For safety reasons, the input voltage was limited to less than $30V_{ac}$. In turn, it would result in lower power losses due to a lower input voltage. Figure 5.2 shows the measured waveforms of the gate driver supply. The voltage ripple was negligible at these levels meaning that the drivers would have stable source for driving the switches. This circuit cannot be simulated due to the complex circuitry of the SR086 chip.

## 5.2 Simulations of designed circuits

The signals discussed from here on are done in the frequency domain. The frequency response gives a more detailed picture than the time domain equivalent due to the random nature of signals under considerations and the time window limitations of oscilloscopes




(a) $15V_{dc}$ DC Output (b) $22V_{dc}$ DC Output

Figure 5.2: Gate Driver Supply outputs at $27V_{ac}$ input

### 5.2.1 PRBS and PRIS

To generate the PRIS, a short circuit was placed across the output terminals of the main circuit and the switching signal was applied. The signal were then logged in simulation and from the physical circuit. The Power Spectral Density (PSD) of the signals were obtained from the script in Appendix I (Listing I.3). Figure 5.3 shows the simulation outputs. At low frequencies, the PRBS has flat response which comes from the signal holding the same value throughout a sample clock period. In contrast, the PRIS shows a rising response at low frequencies due the impulse decaying over the course of a clock period.

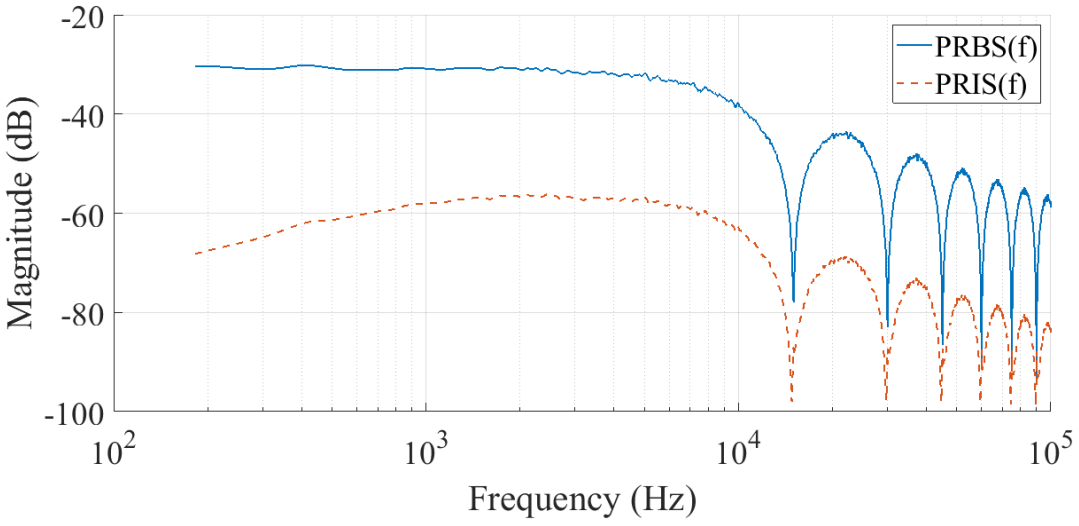



Figure 5.3: Simulation: PRBS and PRIS

Figure 5.4 shows the results obtained from the physical circuit. The PRBS obtained closely resembles the simulation counterpart in terms of low frequency response and decay at multiples of the switching frequency. This means that Listing G.1 is capable of

generating the required switching signal. The measured PRBS also has a similar pattern to the simulated value. However, sampling frequency multiples, its magnitude increases. This is attributed to the non-ideal behaviour of physical components. The switching is not as instantaneous as the simulated components.

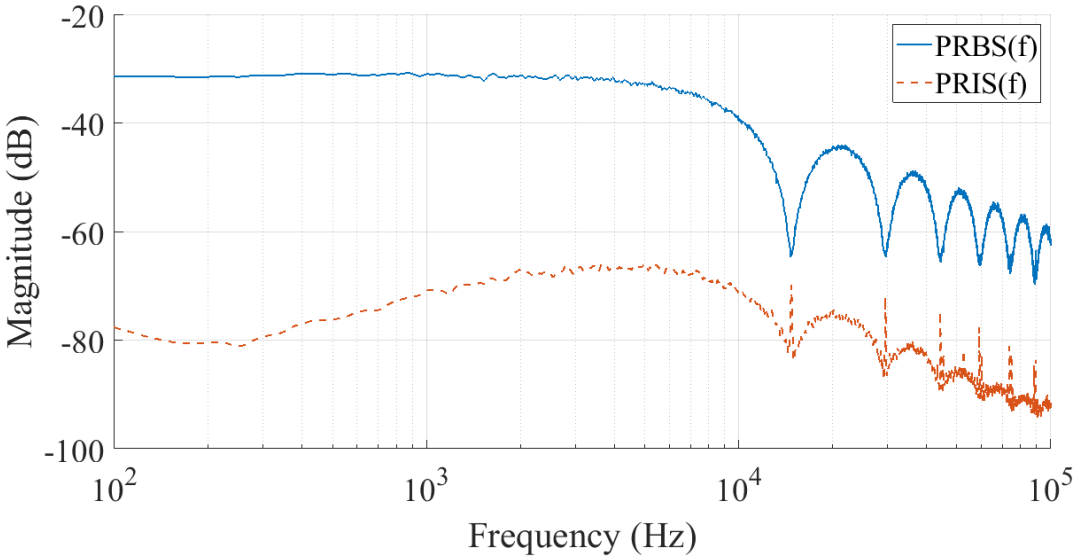



Figure 5.4: Measurement: PRBS and PRIS

### 5.2.2 Control Signals

The functionality of the control signals was verified by measuring the voltage across collector-emitter junction of switch S1 simultaneously with the collector-emitter junction of another switch. S1 and S2 should have the opposite signal as they are both high side switches, S1 and S3 also have opposite signals because they form a half-bridge pair while S1 and S4 have the same signals as they close the circuit. The results are documented in Appendix F.1.

## 5.3 System identification results (Off-line)

To identify a given system, a PRBS signal was sent to the switches in order to generate the current impulses. The results were measured on the DUT and captured in .csv format. In MATLAB, the data was transferred from *Similink* to the workspace. The tests were run for a 2 second duration as this was long enough for the PRBS14 to repeat at 15 kHz. The oscilloscope was set to measure 1.25M points as large datasets give better resolution in the frequency domain. The data was analysed using Listing I.2 in Appendix I

### 5.3.1 Resistive Load

A known system was used as benchmark for the physical measurements. A 100Ω film resistor (LTO150F100R0JTE3), similar to the one used on the main circuit was used as the load due to its high precision value and low susceptibility to line inductance. The resistor model in MATLAB is ideal.

#### Voltage and Current PSD

Figure 5.5 and Figure 5.6 show load voltage ($V_{DUT}(f)$) and load current ($I_{DUT}(f)$) across the observed spectrum. The voltage and current exhibit similar waveforms due to high precision resistors being linear across the frequency spectrum.

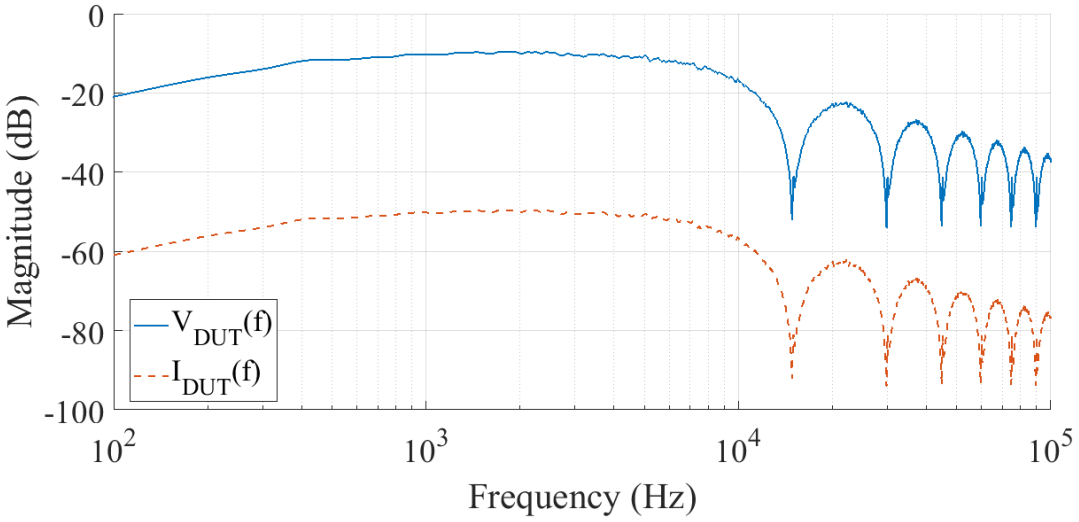



Figure 5.5: Resistor Simulation

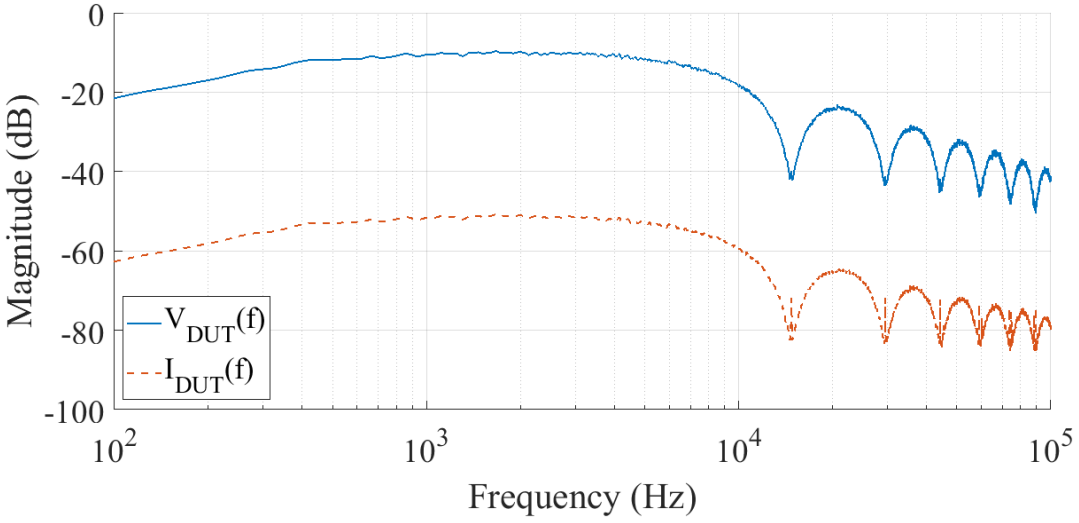



Figure 5.6: Resistor Measurement

### Impedance

The expected value for a 100Ω resistor as a function of frequency 40 dB ($20log_{10}(100) = 40dB$). This has to remain near constant throughout the plot. As expected, in Figure 5.7, the simulation value maintains that value due to the ideal behaviour of the simulator. The

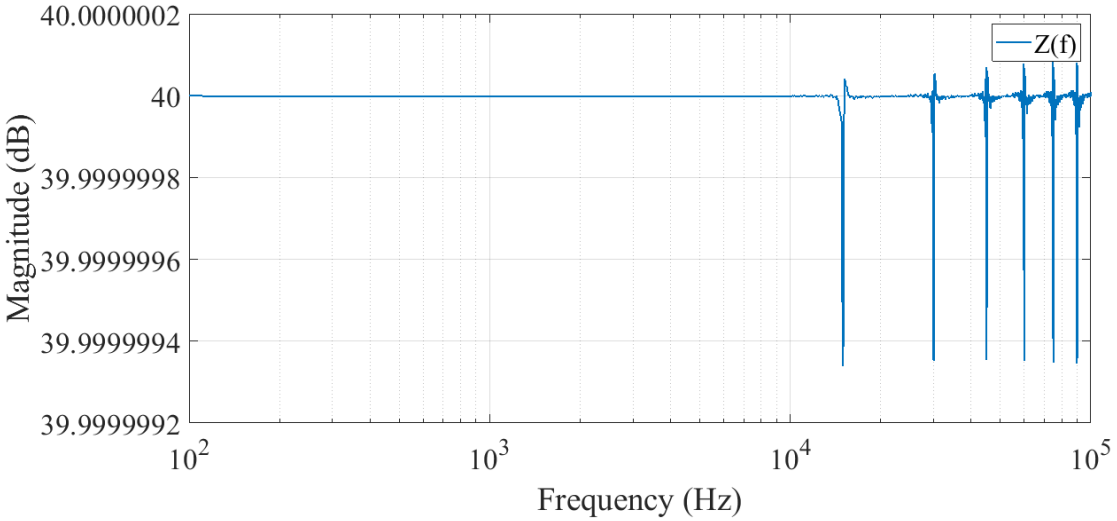



Figure 5.7: Resistor Impedance Simulation

impedance obtained from the measurements in Figure 5.8 shows a flat response across the frequency spectrum. The impedance was measured to be 41.2 dB. Compared to the simulation, the measured value has a constant 1.2 dB difference throughout. The difference is a result of the current probe not being at a true zero value.

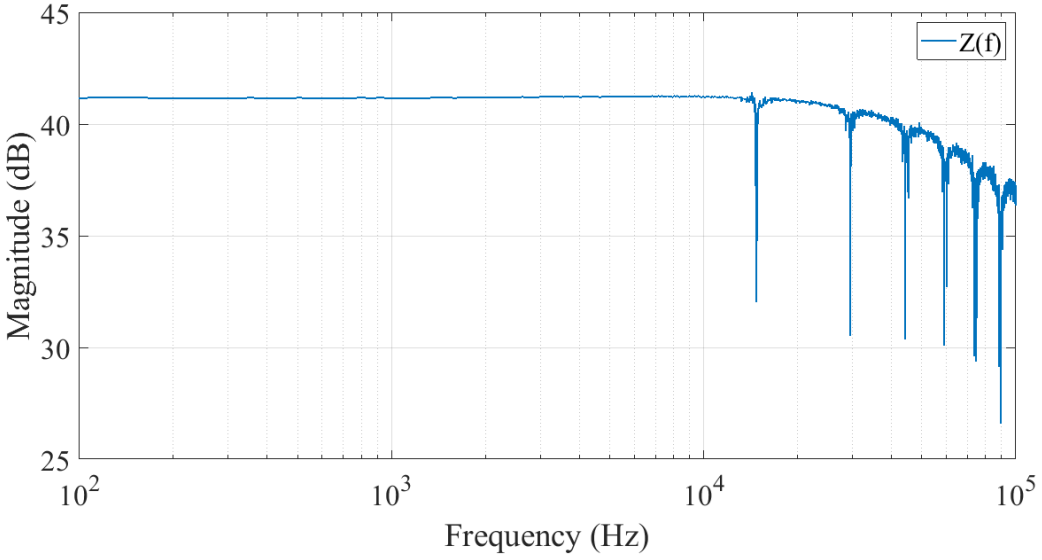



Figure 5.8: Resistance Impedance Measurement

Overall, the observed values indicate that the designed excitation signal and circuit are able to identify a given power system as set out in the objectives.

### 5.3.2 Rheostat Load

To test the applicability of the circuit, other systems were analysed. The systems tested were limited to what was available in the Machines Lab. Rheostats are commonly used as variable high power resistors in the labs. However on closer inspection, it was seen that the rheostats consists of wire windings and a washer that adjusts the output resistance measured. The windings indicate that the device inductance will be very. An LCR meter was used to estimate the LC parameter so that they could be modelled in *Simulink*. The values were obtained at 1 kHz. L was found to be $21.88mH$ and C was $338.3nF$. The equivalent circuit of a rheostat is shown in Figure 5.9

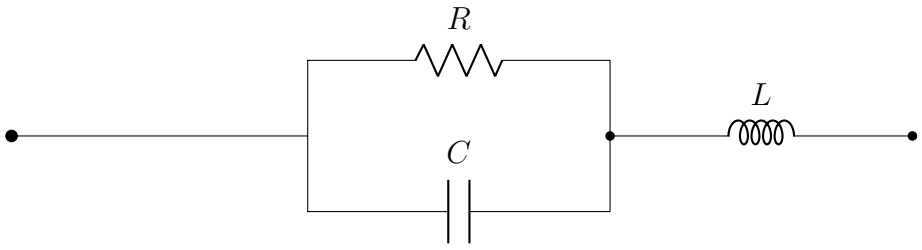



Figure 5.9: Rheostat Equivalent Circuit

The measured waveforms obtained for the rheostat load in Appendix F.2.1 do confirm the high internal inductance as can be seen by the increasing impedance as a function of frequency. The simulated model also shows an increasing impedance as a function of frequency. However the magnitude of differ greatly meaning that the model obtained from the LCR was inaccurate.

### 5.3.3 Rheostat-Inductive (RL) Load

Refer to Appendix F.2.2. The inductor had a measured value of $165mH$.

### 5.3.4 Rheostat-Capacitive (RC) Load

Refer to Appendix F.2.3. The capacitor had a measured value of $50\mu F$.

### 5.3.5 Rheostat-Inductive-Capacitive (RLC) Load

Refer to Appendix F.2.4. The inductor and capacitor used in the RL and RC loads respectively, were used in the RLC load.

### 5.3.6 Summary of system identification

The impedance data collected from simulations and data is summarized in Table 5.3. For a resistive load, the measurements and the simulation data has a notable correlation which reinforces the suitability of the designed circuit and signal.

| | **Frequency** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 100Hz | | 1kHz | | 10kHz | | 20kHz | |
| **Load** | Sim | Meas. | Sim | Meas. | Sim | Meas. | Sim | Meas. |
| Resistor | 40.0 | 41.2 | 40.0 | 41.2 | 40.0 | 41.2 | 40.0 | 41.1 |
| Rheostat | 57.2 | 43.6 | 46.0 | 47.1 | 72.2 | 53.8 | 78.6 | 56.8 |
| Rheostat + Inductor | 57.1 | 43.8 | 46.1 | 47.2 | 72.3 | 53.8 | 78.6 | 55.0 |
| Rheostat + Capacitor | 57.8 | 43.8 | 45.8 | 47.3 | 72.3 | 54.1 | 78.6 | 57.0 |
| Rheostat + Inductor + Capacitor | 57.8 | 44.2 | 45.8 | 47.2 | 72.3 | 52.6 | 78.6 | 53.3 |

Table 5.3: System identification impedance in dB

Based on the waveforms of the rheostat coupled with different loads, it was seen that the addition of the capacitor and inductor did not affect the response if the rheostat. This does make sense given how high the rheostat inductance was relative to the external inductance. The impedance equation of an inductor ($X_L = j\omega L$) shows that in a series connection, the higher inductance is the one that has a major influence on the circuit impedance. When the capacitor is connected to the rheostat in series, the equivalent capacitance will decrease from first principles. From the capacitance impedance equation ($X_C = \frac{1}{j\omega C}$), it can be seen that the capacitive impedance in the load will decrease with an increase in frequency. The leads to all the measured rheostat impedance values increasing due to inductance. This is also shown in Table 5.3 by the magnitude of the impedance, where a rheostat is connected, having similar values at high frequencies. Improving the equivalent parameters of rheostat model will lead to improved simulations results which will give a more meaningful comparison.

# Chapter 6

# Conclusion and Recommendations

## 6.1 Conclusion

The objectives set out in the project comprised of sequential steps from generating a PRBS to ultimately identifying a given power system. The software for the switching control was verified through PSD plots. The generated PRBS signal was then used to control the switches in an H-bridge RLC circuit to generate a bipolar PRIS. The obtained PRIS was verified the same way as PRBS, using PSD plots.

The next step was to verify the functionality of the circuit by using it to identify an known system based on the output voltage and output current. The division of the voltage and current stems directly from Ohm's law but in the frequency domain. This was done partly to prevent dividing zero during the switching interval but mainly to observe how the system responded across the observation frequency spectrum. The identification of a high precision film resistor through simulation (Figure 5.7) and measurements (Figure 5.8) did conform that the signal generator was capable of identifying a system even though it was off-line. The system identification that the circuit gives is only and indicator of how the device under behaves over a frequency range. It does not calculate the value of the parameters as that it subject to the internals of the given device.

The accuracy of the circuit is affected by the equipment used. It was observed that the zero level of the current probe shifted every time measurements were taken. This due to the Hall sensors used in current probes reacting to the pressure at which the contacts touch. The range of the system is currently limited between 100 Hz and 50 kHz due to measurement equipment. The implication is that a limited set of sample frequencies can be used to excite a system.

Overall, the signal generator does meet its set objectives with room for improvement.

## 6.2 Recommendations

When the signal generator is used at higher voltages, clamp snubbers need to be used in order to prevent voltage spikes across the collector-emitter junction. The PCB circuit currently has a slots for snubbers. They sizing of the components is subject to the operational voltages. The input resistor will require a heat sink at higher current outputs as it passes through higher DC currents compared to other components in the circuit. The measurement equipment requires a larger bandwidth compared to the observation spectrum. This will lead to high fidelity measurements. A current sensor can used to improve on the current probe.

# Bibliography


[1] S. Neshvad, S. Chatzinotas, and J. Sachau, "Online determination of grid impedance spectrum through pseudo-random excitation of a pulse width modulator," *Renewable Energy & Power Quality Journal*, 2014.

[2] J. Sun, "Impedance-based stability criterion for grid-connected inverters," *IEEE Transactions on Power Electronics*, vol. 26, no. 11, pp. 3075–3078, Nov 2011.

[3] M. Liserre, R. Teodorescu, and F. Blaabjerg, "Stability of photovoltaic and wind turbine grid-connected inverters for a large set of grid impedance values," *IEEE Transactions on Power Electronics*, vol. 21, no. 1, pp. 263–272, Jan 2006.

[4] M. Cespedes and J. Sun, "Adaptive control of grid-connected inverters based on online grid impedance measurements," *IEEE Transactions on Sustainable Energy*, vol. 5, no. 2, pp. 516–523, April 2014.

[5] P. E. Wellstead, "Non-parametric methods of system identification," *Automatica*, vol. 17, no. 1, pp. 55 – 69, 1981. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0005109881900844

[6] P. M. Nirgude, B. Gunasekaran, Channakeshava, A. D. Rajkumar, and B. P. Singh, "Frequency response analysis approach for condition monitoring of transformer," in *The 17th Annual Meeting of the IEEE Lasers and Electro-Optics Society, 2004. LEOS 2004.*, Oct 2004, pp. 186–189.

[7] A. Abu-Siada, N. Hashemnia, S. Islam, and M. A. S. Masoum, "Understanding power transformer frequency response analysis signatures," *IEEE Electrical Insulation Magazine*, vol. 29, no. 3, pp. 48–56, May 2013.

[8] E. Gomez-Luna, G. A. Mayor, C. Gonzalez-Garcia, and J. P. Guerra, "Current status and future trends in frequency-response analysis with a transformer in service," *IEEE Transactions on Power Delivery*, vol. 28, no. 2, pp. 1024–1031, April 2013.

[9] R. Aarts, "System identification and parameter estimation," *Lecture Notes (University of Twente, 2012). https://www. utwente. nl/ctw/wa/web_dev/old/lectures/113170/notes/notes. pdf*, 2012.

[10] R. D. Nowak, "Nonlinear system identification," *Circuits, Systems and Signal Processing*, vol. 21, no. 1, pp. 109–122, Jan 2002. [Online]. Available: https://doi.org/10.1007/BF01211655

[11] O. Nelles, *Introduction.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 1–19. [Online]. Available: https://doi.org/10.1007/978-3-662-04323-3_1

[12] R. Haber, "Nonlinearity tests for dynamic processes," *IFAC Proceedings Volumes*, vol. 18, no. 5, pp. 409 – 414, 1985, 7th IFAC/IFORS Symposium on Identification and System Parameter Estimation, York, UK, 3-7 July. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667017605949

[13] B. Mettler, *Frequency Response System Identification.* Boston, MA: Springer US, 2003, pp. 29–51. [Online]. Available: https://doi.org/10.1007/978-1-4757-3785-1_2

[14] S. A. Ryder, "Diagnosing transformer faults using frequency response analysis," *IEEE Electrical Insulation Magazine*, vol. 19, no. 2, pp. 16–22, March 2003.

[15] A. Keyhani, H. Tsai, S. Pillutla, and A. Abur, "Identification of high frequency transformer model parameters," *Electric Power Systems Research*, vol. 42, no. 2, pp. 127 – 133, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378779696011972

[16] G. M. V. Zambrano, A. C. Ferreira, and L. P. Caloba, "Power transformer equivalent circuit identification by artificial neural network using frequency response analysis," in *2006 IEEE Power Engineering Society General Meeting*, June 2006, pp. 6 pp.–.

[17] S. M. Islam, K. M. Coates, and G. Ledwich, "Identification of high frequency transformer equivalent circuit using matlab from frequency domain data," in *IAS '97. Conference Record of the 1997 IEEE Industry Applications Conference Thirty-Second IAS Annual Meeting*, vol. 1, Oct 1997, pp. 357–364 vol.1.

[18] R. R. Pintelon, *System identification : a frequency domain approach.* New York: IEEE Press, 2001.

[19] K. Godfrey, Ed., *Perturbation Signals for System Identification.* Hertfordshire, UK, UK: Prentice Hall International (UK) Ltd., 1993.

[20] D. Brown, G. Carbon, and K. Ramsey, "Survey of excitation techniques applicable to the testing of automotive structures," in *1977 International Automotive Engineering Congress and Exposition.* SAE International, feb 1977. [Online]. Available: https://doi.org/10.4271/770029

[21] K. Godfrey, *Perturbation signals for system identification* . New York : Prentice Hall, 1993, includes bibliographical references (p. 422-435) and index.

[22] M. Allain, P. Viarouge, and F. Tourkhani, "The use of pseudo-random binary sequences to predict a dc-dc converter's control-to-ouput transfer function in continuous conduction mode," in *Canadian Conference on Electrical and Computer Engineering, 2005.*, May 2005, pp. 574–577.

[23] H. Okawara, "Hideo Okawara's Mixed Signal Series DSP-Based Testing-Fundamentals 50 PRBS (Pseudo Random Binary Sequence)," *Advantest Corporation*, 2013.

[24] K. R. Godfrey, H. A. Barker, and A. J. Tucker, "Comparison of perturbation signals for linear system identification in the frequency domain," *IEE Proceedings - Control Theory and Applications*, vol. 146, no. 6, pp. 535–548, Nov 1999.

[25] T. Roinila, M. Vilkko, and J. Sun, "Broadband methods for online grid impedance measurement," in *2013 IEEE Energy Conversion Congress and Exposition*, Sept 2013, pp. 3003–3010.

[26] B. P. Lathi, *Modern Digital and Analog Communication Systems*, 4th ed. New York: Oxford University Press, 2010.

[27] K. Schon, *Characterisation and Generation of High Impulse Voltages and Currents*. Heidelberg: Springer International Publishing, 2013, pp. 5–38. [Online]. Available: https://doi.org/10.1007/978-3-319-00378-8_2

[28] V. Cooray, *Lightning Protection*, ser. IET power and energy series ; 58. Stevenage: IET, 2010.

[29] IEEE, "IEEE Standard for High-Voltage Testing Techniques," *IEEE Std 4-2013 (Revision of IEEE Std 4-1995)*, pp. 1–213, May 2013.

[30] D. G. Alciatore, *Introduction to mechatronics and measurement systems*. Tata McGraw-Hill Education, 2007.

[31] *High Voltage IGBT IXGH24N170*, IXYS CORPORATION, September 2008. [Online]. Available: http://ixapps.ixys.com/DataSheet/DS98994A(IXGH-T24N170).pdf

[32] S. Sapre, "Isolated gate drivers-what, why, and how?" *Analog Dialogue*, vol. 52, no. 2, June 2018. [Online]. Available: https://www.analog.com/media/en/analog-dialogue/volume-52/number-2/isolated-gate-drivers-what-why-and-how.pdf

[33] *AN486: High-Side Bootstrap Design Using ISODrivers in Power Delivery Systems*, Silicon Labs, rev. 0.2. [Online]. Available: https://www.silabs.com/documents/public/application-notes/AN486.pdf

[34] *0.5 and 4.0 Amp ISOdrivers (2.5 and 5 kVRMS)*, Silicon Labs, rev. 2.13. [Online]. Available: https://www.silabs.com/documents/public/data-sheets/Si823x.pdf

[35] *Adjustable Off-Line Inductorless Switching Regulator*, Microchip, revision A (May 2017). [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/20005544A.pdf

[36] "IPC-2152 - Standard for Determining Current-Carrying Capacity in Printed Board Design," *Circuit World*, vol. 36, no. 1, p. null, 2010. [Online]. Available: https://doi.org/10.1108/cw.2010.21736aab.003

[37] PortManipulation. Arduino. [Online]. Available: https://www.arduino.cc/en/Reference/PortManipulation

[38] A. Klein, *Linear Feedback Shift Registers*. London: Springer London, 2013, pp. 17–58. [Online]. Available: https://doi.org/10.1007/978-1-4471-5079-4_2

# Appendices

# Appendix A

# Project Planning Schedule

## A.1 Planned Schedule

| Week | Dates | Scheduled Tasks |
|---|---|---|
| | | LITERATURE REVIEW |
| 1 | 23/07 - 29/07 | System identification overview |
| 2 | 30/07 - 05/08 | Excitation/Perturbation signals |
| 3 | 06/08 - 12/08 | Fundamentals impulse generation |
| | | SOFTWARE DESIGN |
| 4 | 13/08 - 19/08 | Designing PRBS signal |
| 5 | 20/08 - 26/08 | Circuit simulations |
| 6 | 27/08 - 02/09 | Component sizing |
| | | HARDWARE DESIGN |
| 7 | 03/09 - 09/09 | Schematic design of circuits |
| 8 | 10/09 - 16/09 | Sourcing of components |
| 9 | 17/09 - 23/09 | Manufacturing of PCB (Report writing) |
| | | BUILDING AND TESTING |
| 10 | 01/10 - 07/10 | Testing of power supplies (Report writing) |
| 11 | 08/10 - 14/10 | Testing of main circuit PCB functionality |
| 12 | 15/10 - 21/10 | Power system identification |
| | | REPORT |
| 13 | 22/10 - 28/10 | Revision of system |
| 14 | 29/10 - 04/10 | Finalizing of report |

# Appendix B

# ECSA Outcomes Compliance

| Outcome | Project Reference |
|---|---|
| **Problem solving:** Identify, formulate, analyse and solve complex engineering problems creatively and innovatively. | Generating a signal that was suitable for power system identification based on first principles (Chapter 3.2.1) |
| **Application of scientific and engineering knowledge:** Apply knowledge of mathematics, natural sciences, engineering fundamentals and an engineering speciality to solve complex engineering problems. | Derivation of output equations in order to determine the required RLC component sizes. |
| **Engineering Design:** Perform creative, procedural and non-procedural design and synthesis of components, systems, engineering works, products or processes. | Designing power supplies suitable for the circuit application. Using designed power supplies for use with main circuit. Design of multiple PCBs |
| **Investigations, experiments and data analysis:** Demonstrate competence to design and conduct investigations and experiments. | Chapter 5.3 contains the experiment setup of the project, from which data was collected and analysed based on the primary objectives. Use of MATLAB for data analysis (Appendix I) |
| **Engineering methods, skills and tools, including Information Technology:** Demonstrate competence to use appropriate engineering methods, skills and tools, including those based on information technology. | Performing PSDs (Appendix I), creating a GUI (Appendix H), generating PRBS signal (Appendix G) |
| **Professional and technical communication:** Demonstrate competence to communicate effectively, both orally and in writing, with engineering audiences and the community at large. | Writing a project report to present work done, poster design and oral presentation. |
| **Individual work:** Demonstrate competence to work effectively as an individual. | Extensive literature review (Chapter 2) in order to form a strong subject knowledge base. Circuit designing, simulations and measurements |
| **Independent Learning Ability:** Demonstrate competence to engage in independent learning through welldeveloped learning skills. | Designing a 4-layer PCB based on the circuit requirements |

# Appendix C

# Bill of Materials

| **Type** | **Manufacturer ID** | **Quantity** | **Cost (ZAR)** |
|---|---|---|---|
| Switches | IXGH24N170 | 6 | R 1,587.00 |
| Resistor | LTO150F100R0JTE3 | 4 | R 910.80 |
| Capacitor | C4GAMUD4200AA1J | 5 | R 569.25 |
| Inductor | RS Pro 1048448 | 2 | R 179.88 |
| Isolated gate drivers | SI8233BB-C-IS1 | 5 | R 282.80 |
| IC Adapter | W9503RC | 5 | R 279.85 |
| Bootstrap capacitor | 450PK3R3MEFC8X11.5 | 5 | R 60.70 |
| Bootstrap resistor | PR01000102208JA100 | 10 | R 8.50 |
| Bootstrap diode | UF1007 | 10 | R 79.90 |
| Micro-controller | A000005 | 1 | R 323.77 |

Table C.1: Bill of Materials (RLC)

| **Type** | **Manufacturer ID** | **Quantity** | **Cost (ZAR)** |
|---|---|---|---|
| Regulator IC | SR086SG-G | 5 | R 117.90 |
| IC Adapter | W9501RC | 5 | R 219.75 |
| Switch | STGD5NB120SZT4 | 5 | R 180.85 |
| Bridge rectifier | DF10M | 3 | R 33.48 |
| Varistor | ERZ-V14D391 | 5 | R 33.85 |
| Fuse | 0034.3118 | 5 | R 46.50 |
| Resistor | FW30A10R0JA | 10 | R 39.75 |
| Resistor | ROX3SJ10K | 10 | R 11.20 |
| Resistor | ROX3SJ82K | 10 | R 12.70 |
| Resistor | ROX3SJ180K | 10 | R 33.20 |
| Resistor | ROX3SJ1M0 | 10 | R 33.00 |
| Potentiometer | RK11K1140A72 | 3 | R 80.55 |
| Output capacitor | UVZ2A471MHD | 5 | R 75.40 |

Table C.2: Bill of Materials (Gate driver supply)

| Type | Manufacturer ID | Quantity | Cost (ZAR) |
|---|---|---|---|
| Diodes | VS-40EPS12PBF | 6 | R 347.10 |
| Capacitors | ELHS451VSN821MA60S | 2 | R 409.40 |
| Bridge Rectifier | FBO40-12N | 1 | R 147.43 |

Table C.3: Bill of Materials (Full Bridge Rectifier)

| Type | Manufacturer ID | Quantity | Cost (ZAR) |
|---|---|---|---|
| Binding Posts | Keystone Electronics 7006 | 10 | R 1,320.90 |
| Binding Posts | Keystone Electronics 7007 | 10 | R 1,320.90 |
| Hex Nuts | RS Pro 525-701 | 2 | R 151.40 |
| Hex Spacer | HTSN-M3-20-6-2 | 20 | R 104.40 |
| Decoupling capacitor | RCER72A105K2K1H03BÂ | 10 | R 71.40 |
| Decoupling capacitor | C320C104K1R5TAÂ | 10 | R 59.25 |
| Decoupling capacitor | UPW2A010MDDÂ | 10 | R 16.00 |
| Decoupling capacitor | UPW2A100MEDÂ | 10 | R 26.20 |

Table C.4: Bill of Materials (Miscellaneous)

| Type | Manufacturer ID | Quantity | Cost (ZAR) |
|---|---|---|---|
| Main Circuit | JLCPCB | 5 | R 1,580.00 |

Table C.5: Bill of Materials (Printed Circuit Board)

# Appendix D

# Derivations and Calculations

## D.1 RLC Derivations

Mathematical form of a series RLC circuit in Figure 3.3

$$Ri + \frac{1}{C}\int_{t_0}^{t} i(x)dx + v_c(t_0) + L\frac{di}{dt} = v_s(t) \tag{D.1}$$

Taking time derivative and rearranging terms

$$\frac{d^2i}{dt^2} + \frac{R}{L}\frac{di}{dt} + \frac{i}{LC} = \frac{dv_s}{dt} \tag{D.2}$$

Applying the Fourier transform

$$s^2 + \frac{R}{L}s + \frac{i}{LC} = 0 \tag{D.3}$$

The roots to the equation

$$s_{1,2} = -\frac{R}{2L} \pm \sqrt{\left(\frac{R}{2L}\right)^2 - \frac{1}{LC}} \tag{D.4}$$

$$s_{1,2} = -\alpha \pm \sqrt{\alpha^2 - \omega_0^2} \tag{D.5}$$

where

$$\alpha = \frac{R}{2L} \text{ rad/s} \tag{D.6}$$

$$\omega_0 = \frac{1}{\sqrt{LC}} \text{ rad/s} \tag{D.7}$$

## D.2 Bootstrap Calculations

Bootstrap design procedure as taken from [33].

| Parameter | Value |
|---|---|
| IGBT gate charge, $Q_G$ | 106nC |
| $t_{MAX}$ | 933.33$\mu$S |
| $t_{MIN}$ | 66.67$\mu$S |

Table D.1: Bootstrap design specifications

$$\begin{aligned} Q_{CB} &= Q_G + (D \times t_{cyc} + I_B) \\ &= 106nC + (3mA \times 933.33\mu S) \\ &= 2.906\mu C \end{aligned} \tag{D.8}$$

$$\begin{aligned} C_B &\geq \frac{Q_{CB}}{\Delta V_{CB}} \\ &\geq \frac{2.906 \times 10^{-6}}{0.05 \times 18} \\ &\geq 3.229\mu F \end{aligned} \tag{D.9}$$

Take bootstrap capacitor as **3.3**$\mu F$

$$\begin{aligned} R_B &\leq \frac{(1 - D_{max})}{5 \times C_B} \\ &\leq \frac{66.67 \times 10^{-6}}{5 \times 3.3\mu F} \\ &\leq 4.04\Omega \end{aligned} \tag{D.10}$$

Take bootstrap resistor as **2.2**$\Omega$

$$\begin{aligned} I_{CHRG} &= \frac{Q_{CB}}{t_{L(MIN)}} \\ &= \frac{2.906 \times 10^{-6}}{66.67 \times 10^{-6}} \\ &= 42.59mA \end{aligned} \tag{D.11}$$

$$\begin{aligned} I_{RBAV(max)} &= \frac{Q_G}{t_{cyc}} + I_B \\ &= \frac{106 \times 10^{-9}}{66.67 \times 10^{-6}} + 3 \times 10^{-3} \\ &= 4.59mA \end{aligned} \tag{D.12}$$

# Appendix E

# Circuit Schematics and Builds

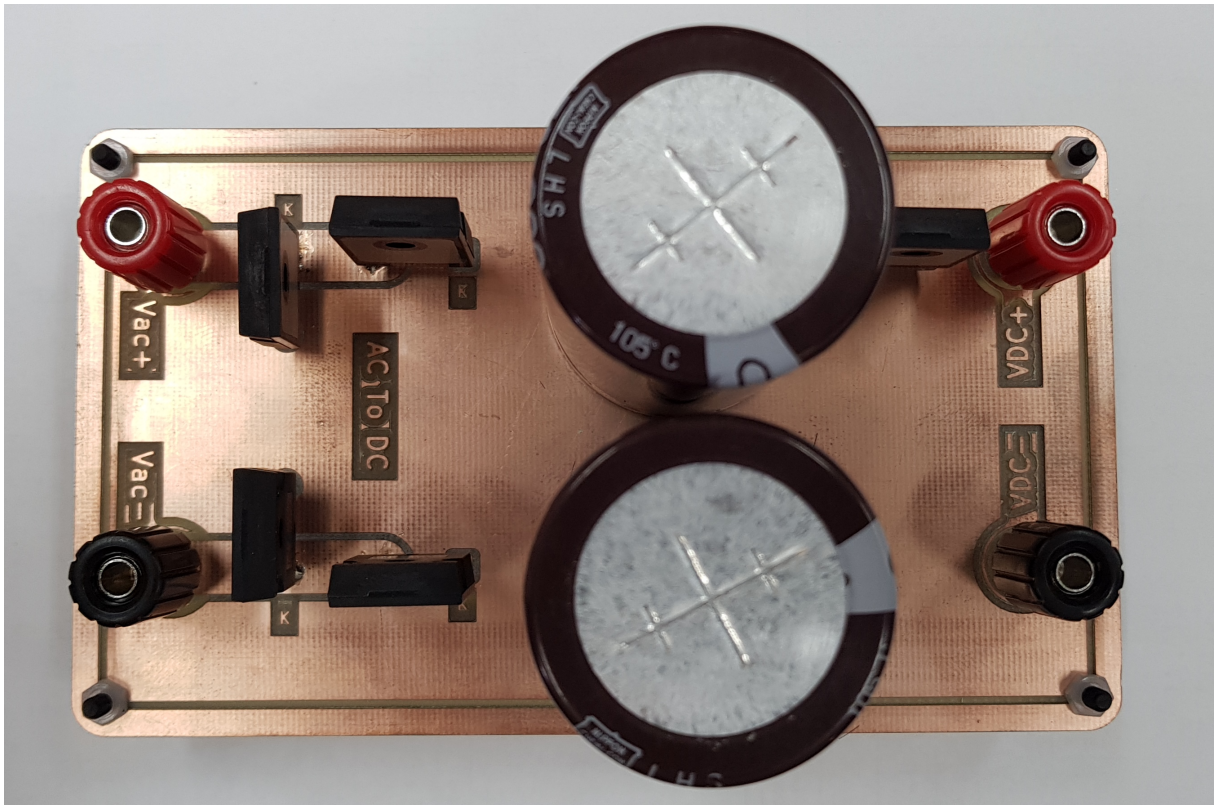



Figure E.1: Full Bridge Rectifier Build

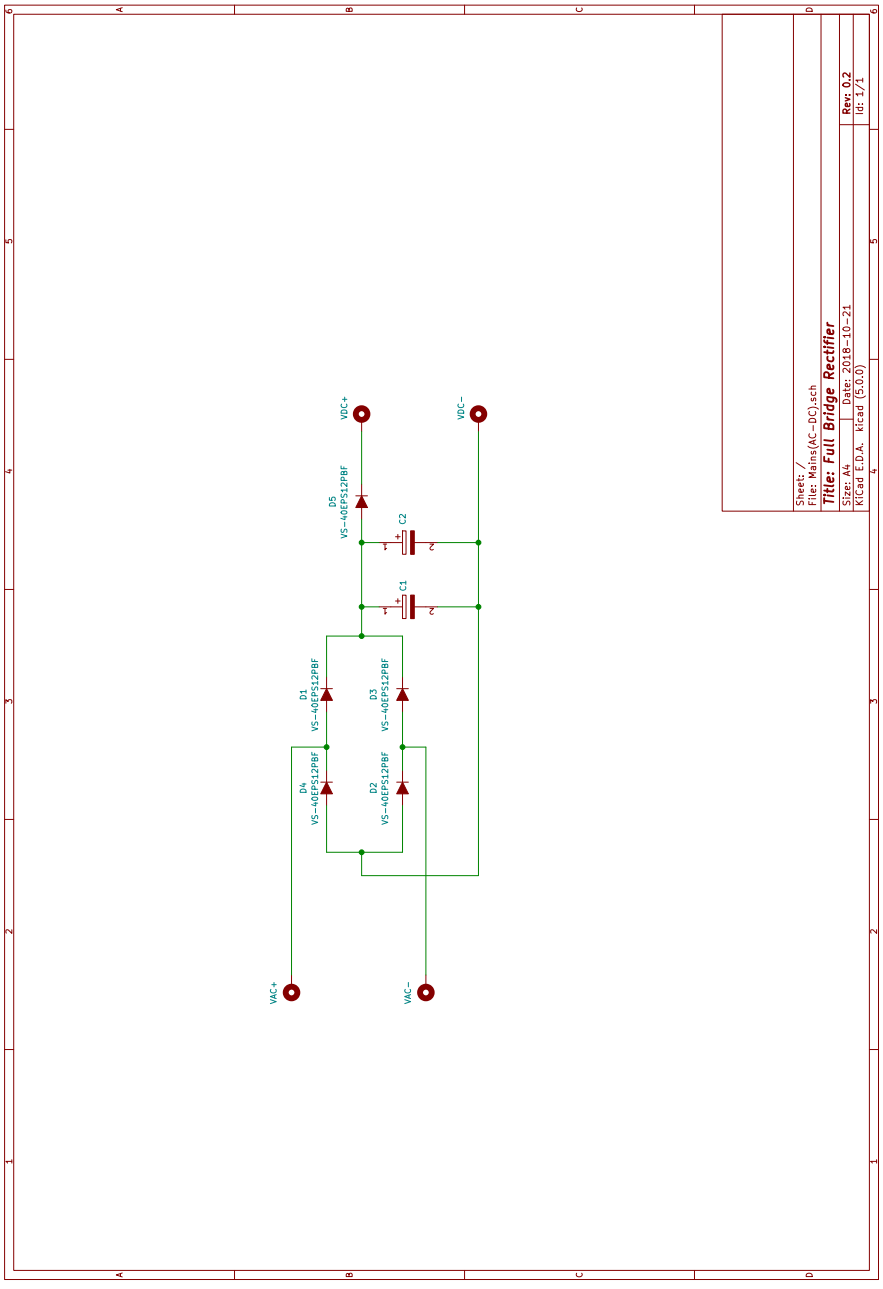



Figure E.2: Full Bridge Rectifier

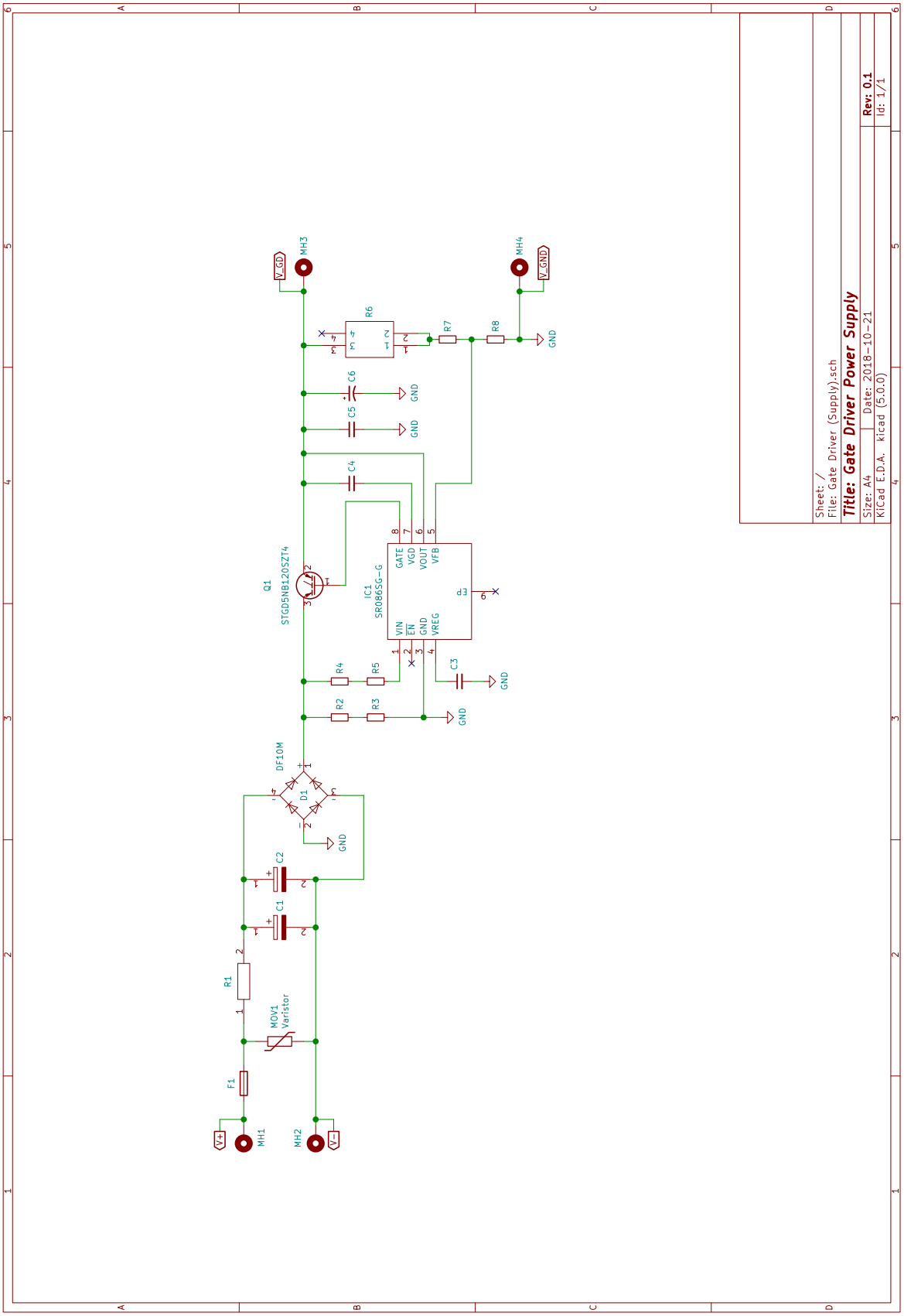



Figure E.3: Gate Driver Power Supply

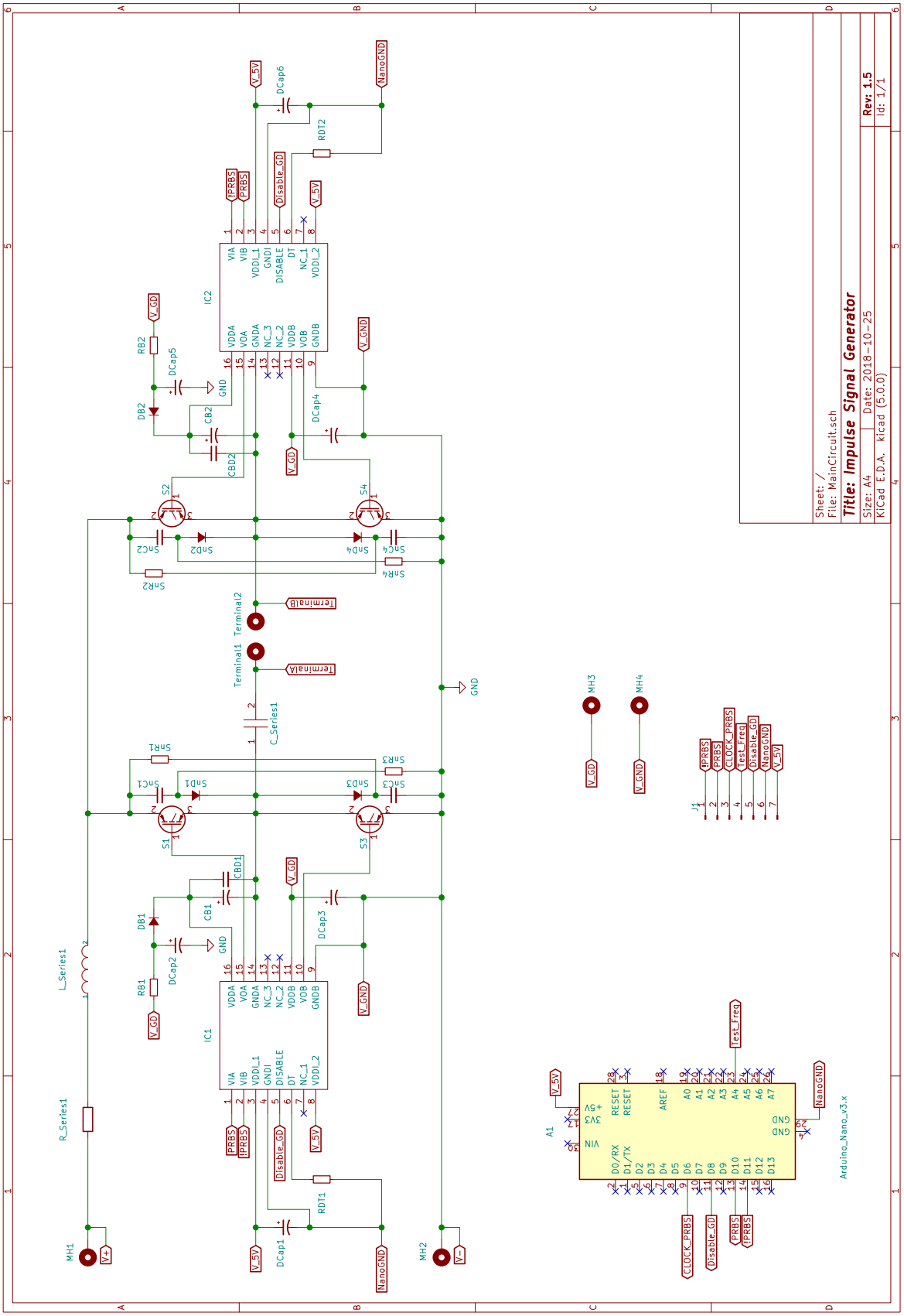



Figure E.4: Impulse Signal Generator

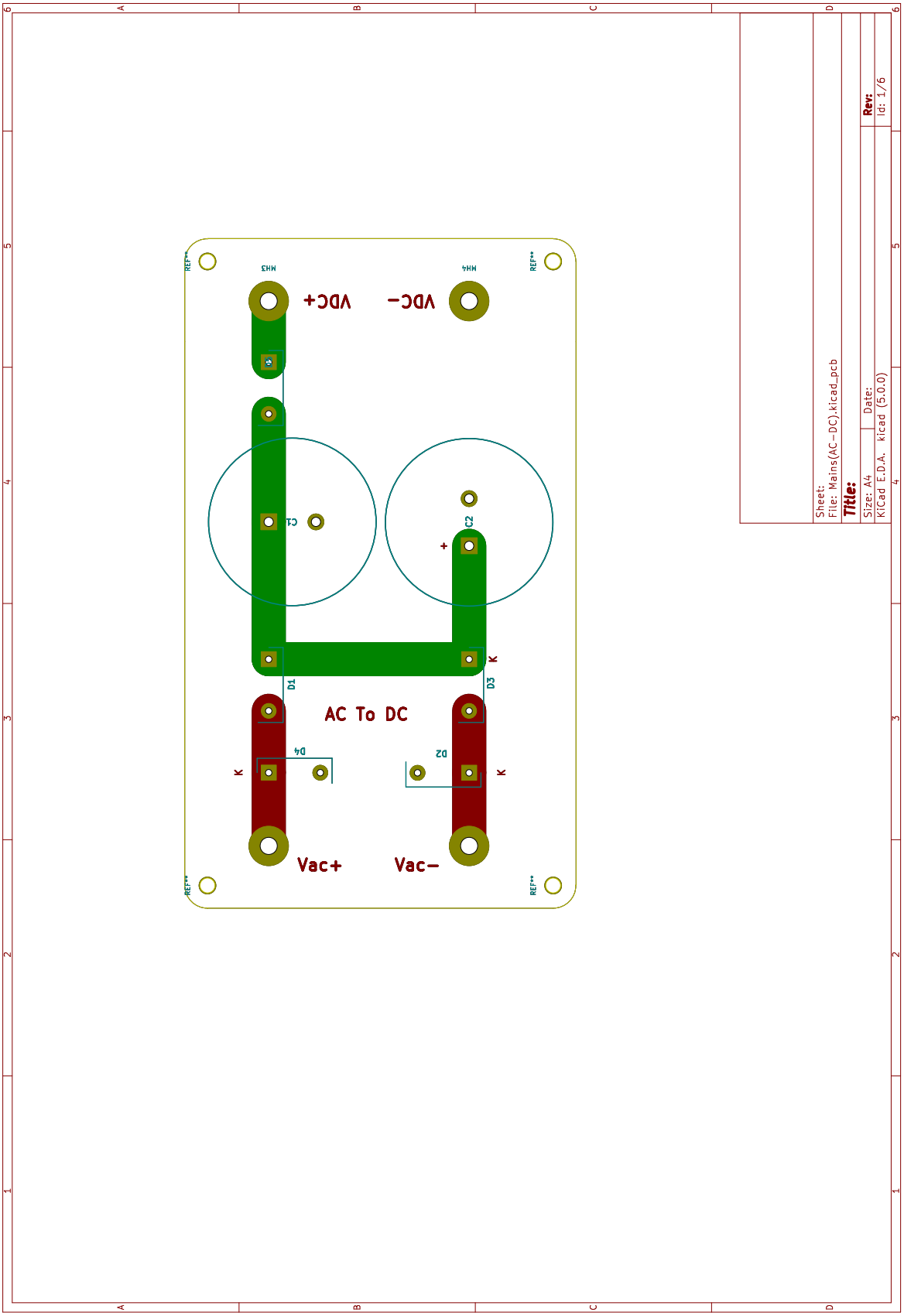



Figure E.5: Full Bridge Rectifier PCB

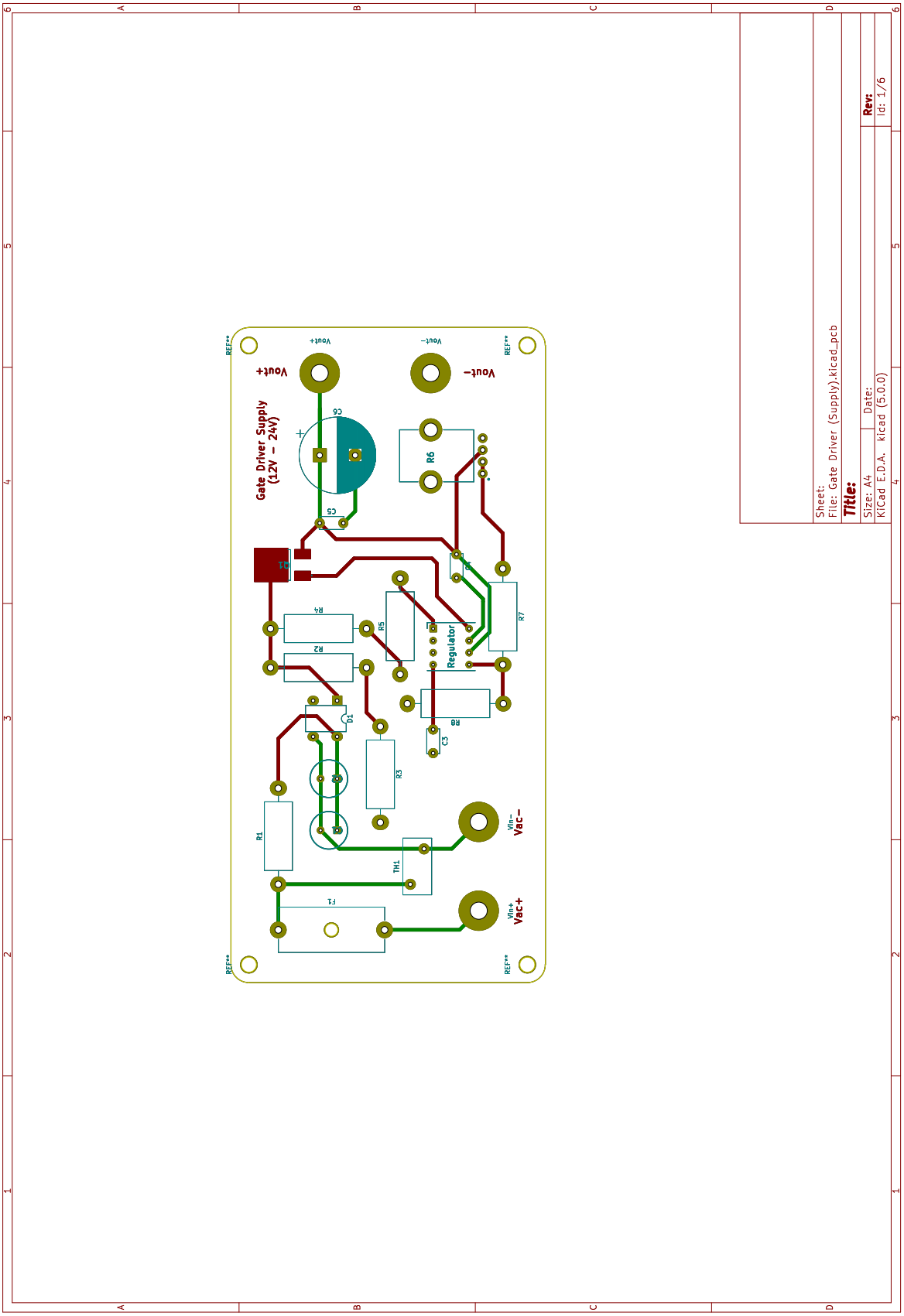



Figure E.6: Gate Driver Power Supply PCB

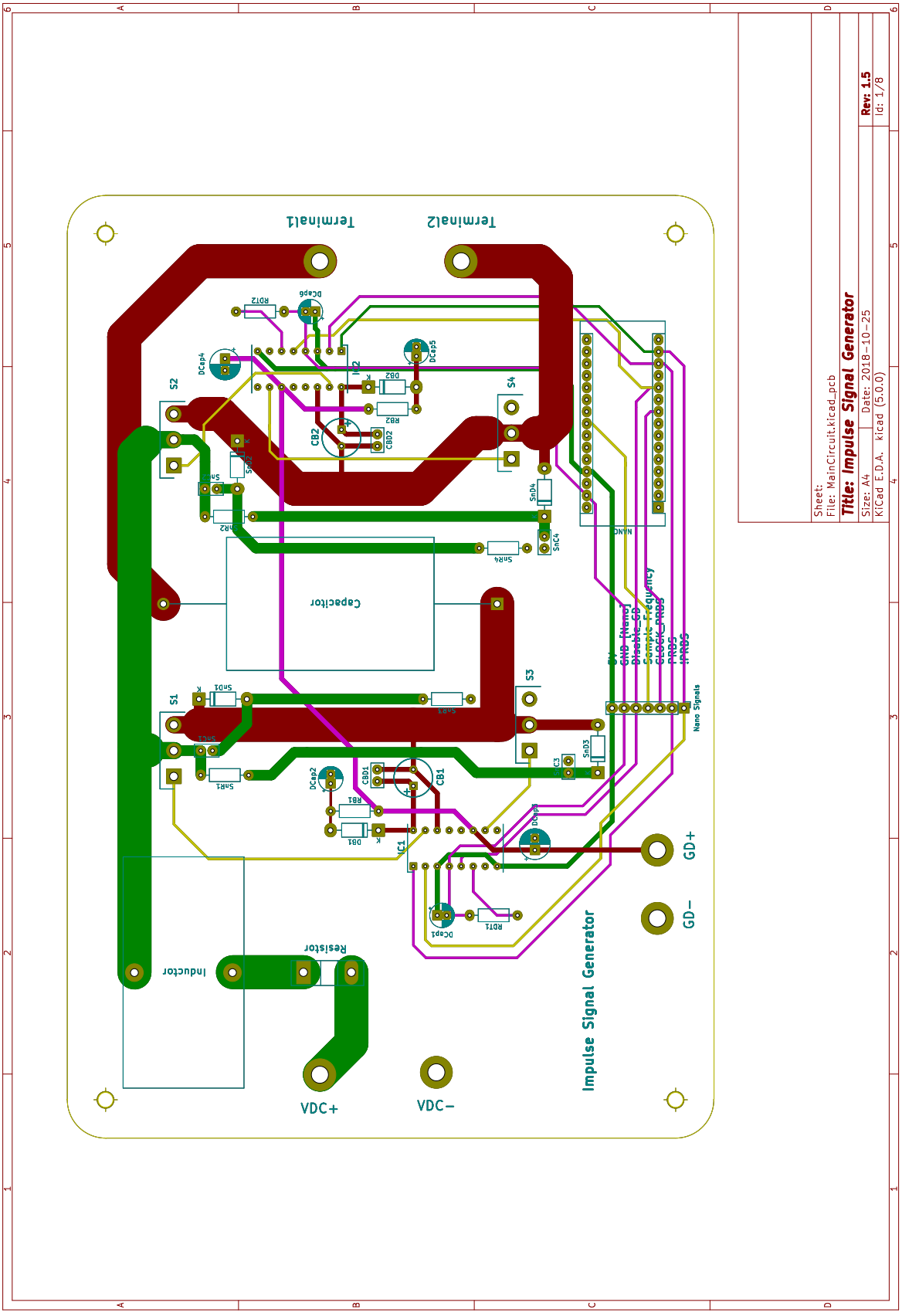



Figure E.7: Impulse Signal Generator PCB

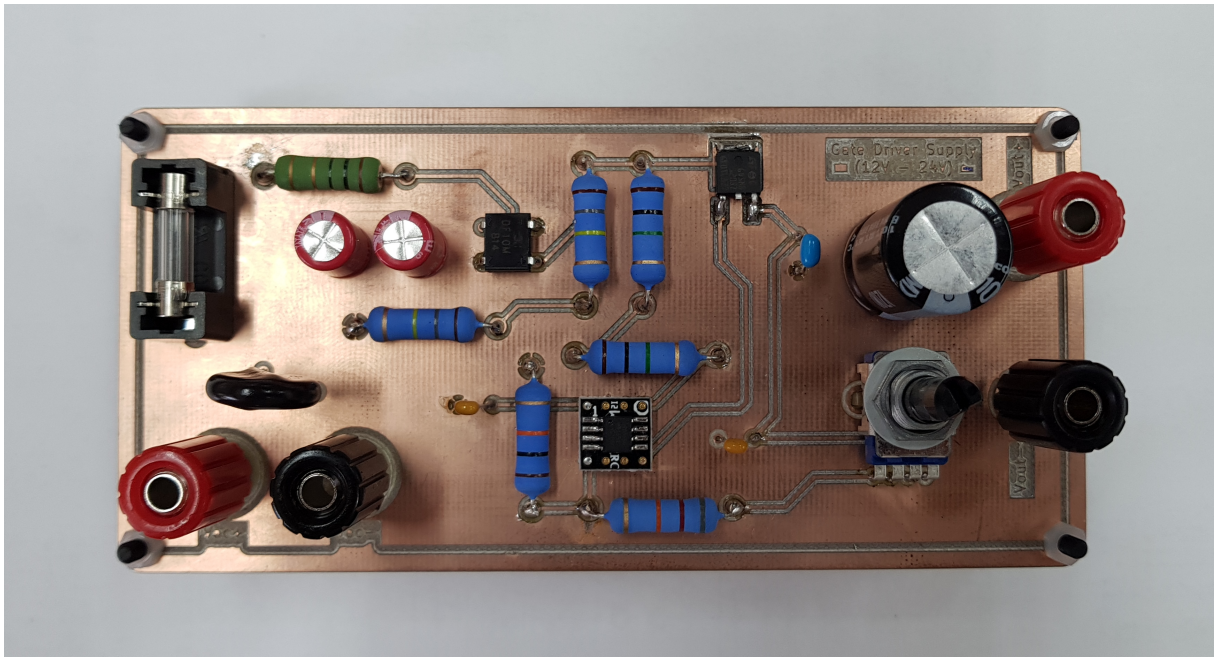



Figure E.8: Gate Driver Power Supply Build

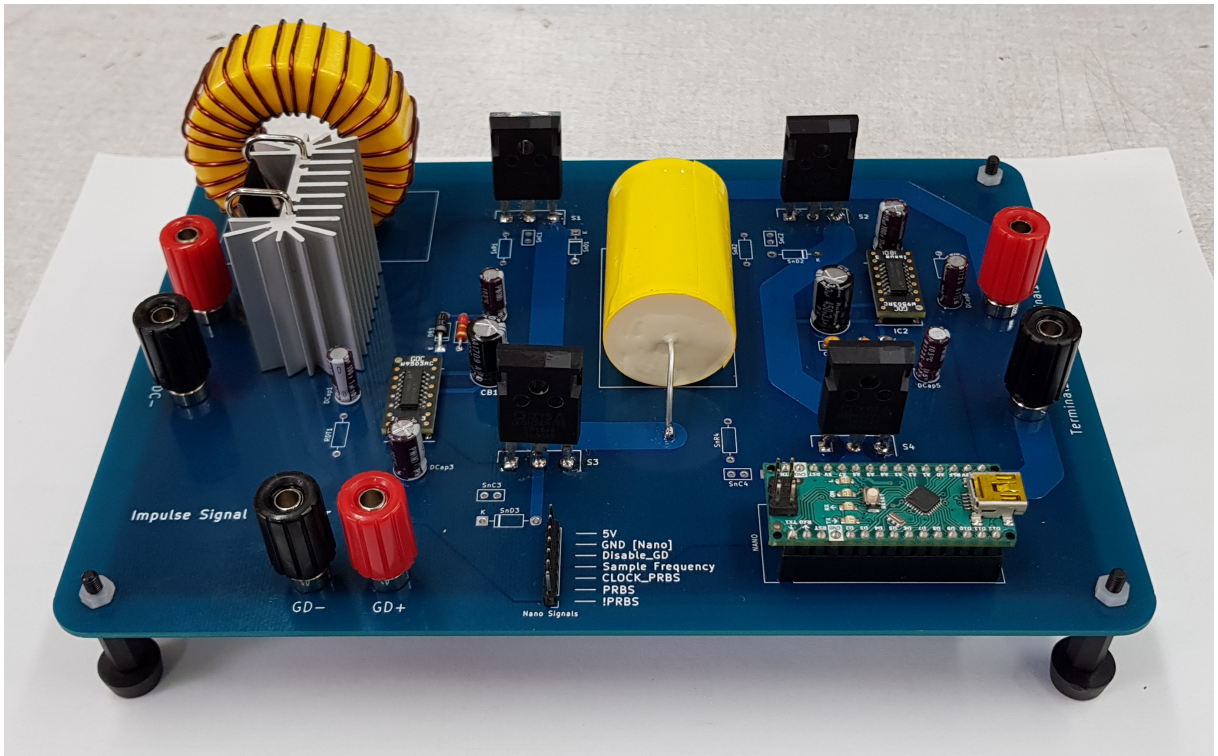



Figure E.9: Impulse Signal Generator Build

# Appendix F

# Results and Measurements

## F.1 Control Signals

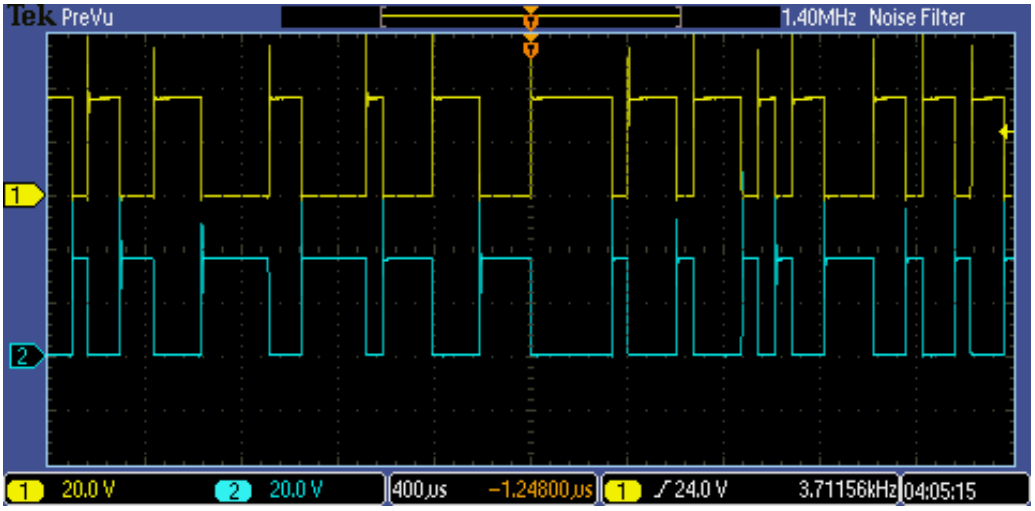



Figure F.1: S1$V_{CE}$ and S2$V_{CE}$

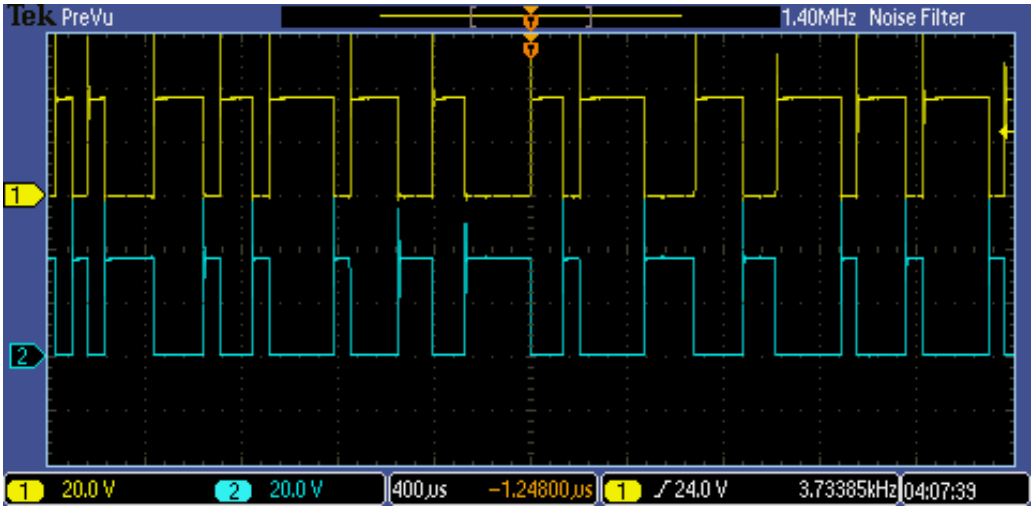



Figure F.2: S1$V_{CE}$ and S3$V_{CE}$

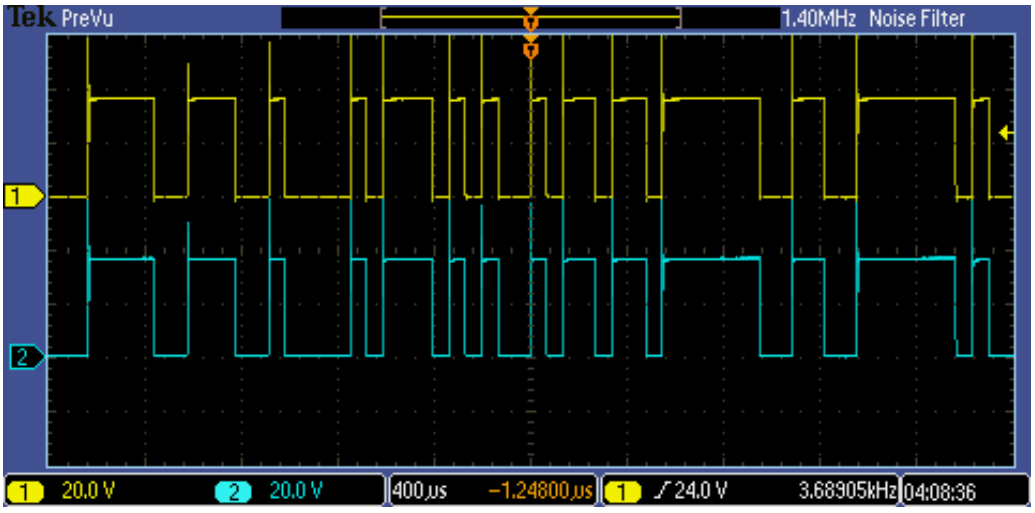



Figure F.3: S1$V_{CE}$ and S4$V_{CE}$

## F.2 System identification results (Off-line)

### F.2.1 Rheostat Load

#### Voltage and Current PSD

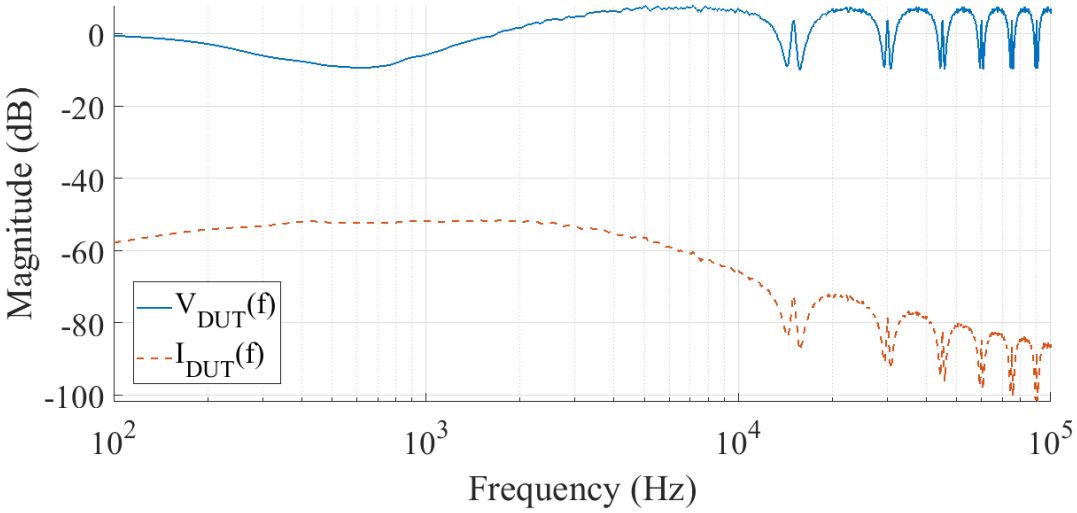



Figure F.4: Rheostat Simulation

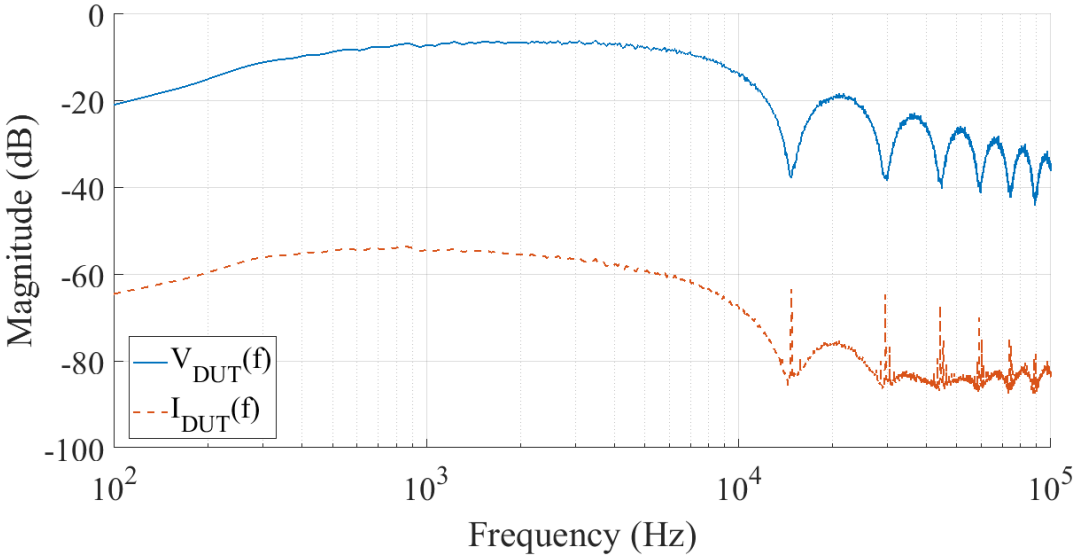



Figure F.5: Rheostat Measurement

### Impedance

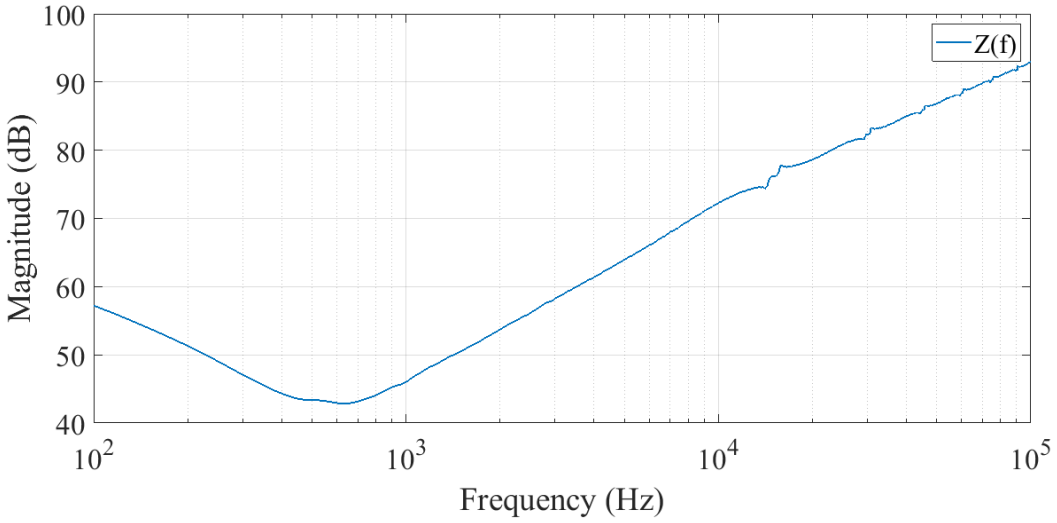



Figure F.6: Rheostat Impedance Simulation

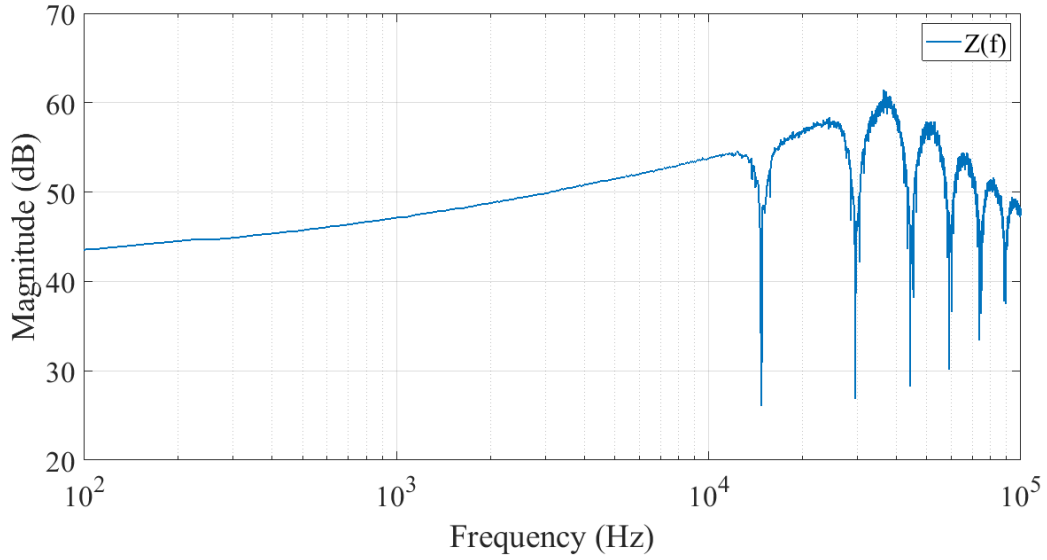



Figure F.7: Rheostat Impedance Measurement

### F.2.2 Rheostat-Inductive (RL) Load

**Voltage and Current PSD**

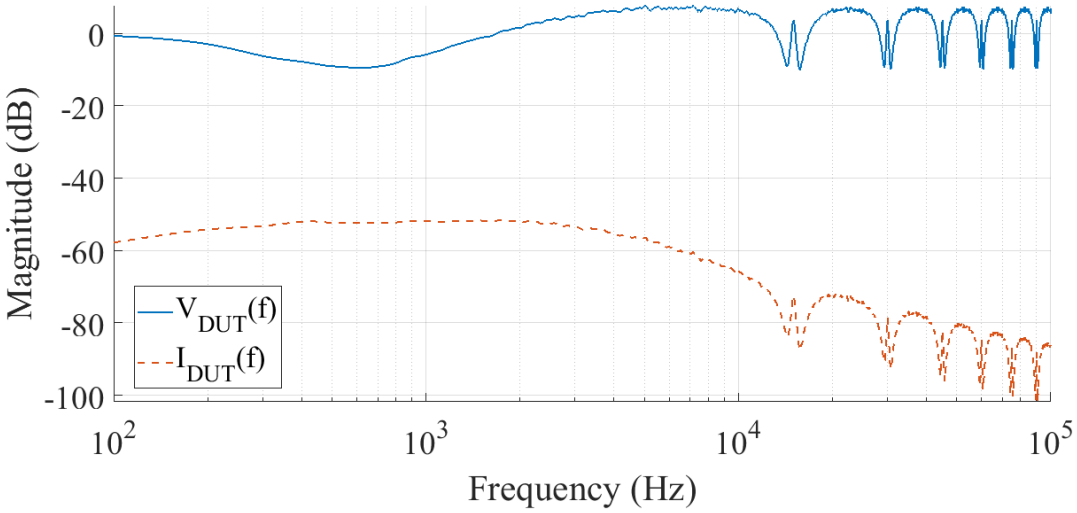



Figure F.8: Rheostat-Inductor Simulation

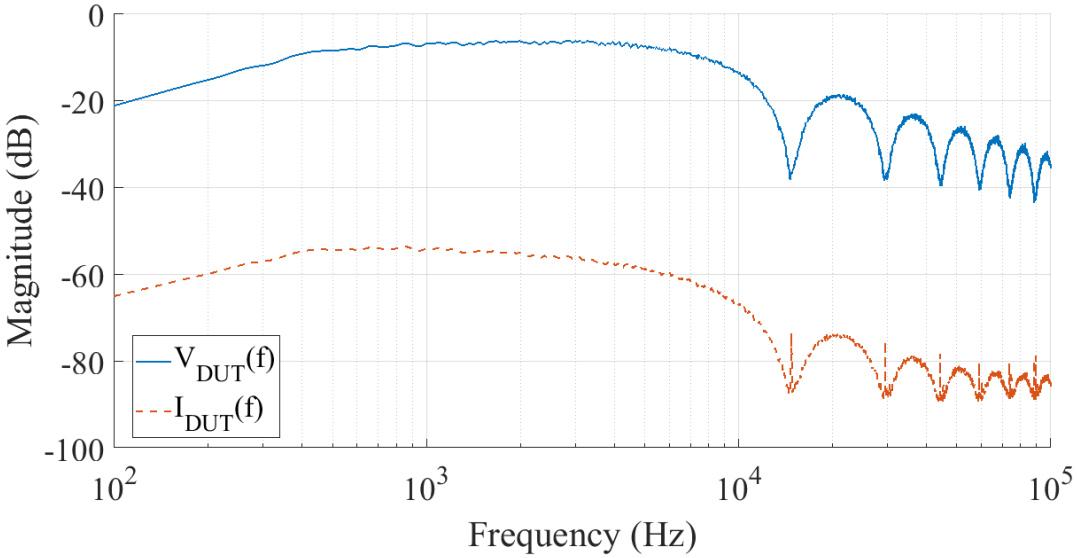



Figure F.9: Rheostat-Inductor Measurement

### Impedance

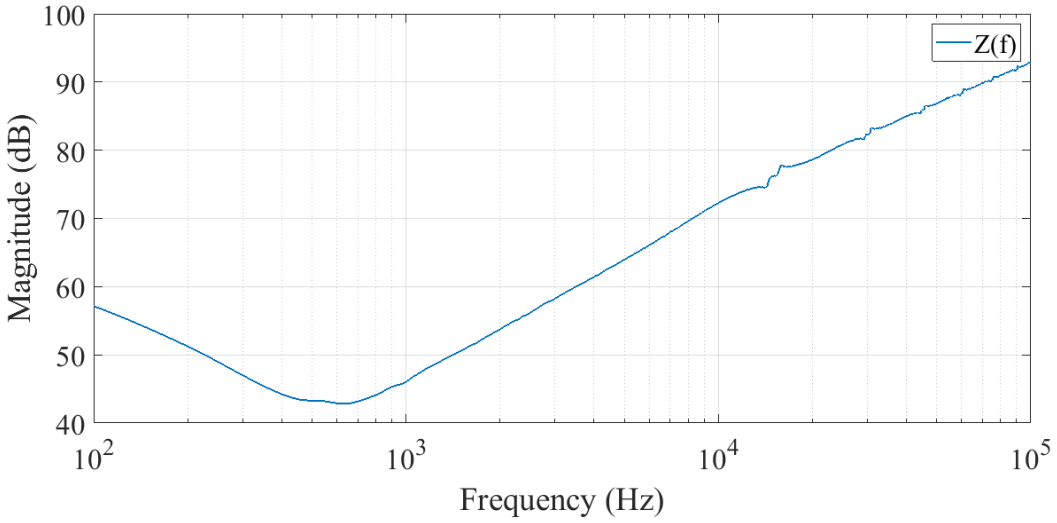



Figure F.10: Rheostat-Inductor Impedance Simulation

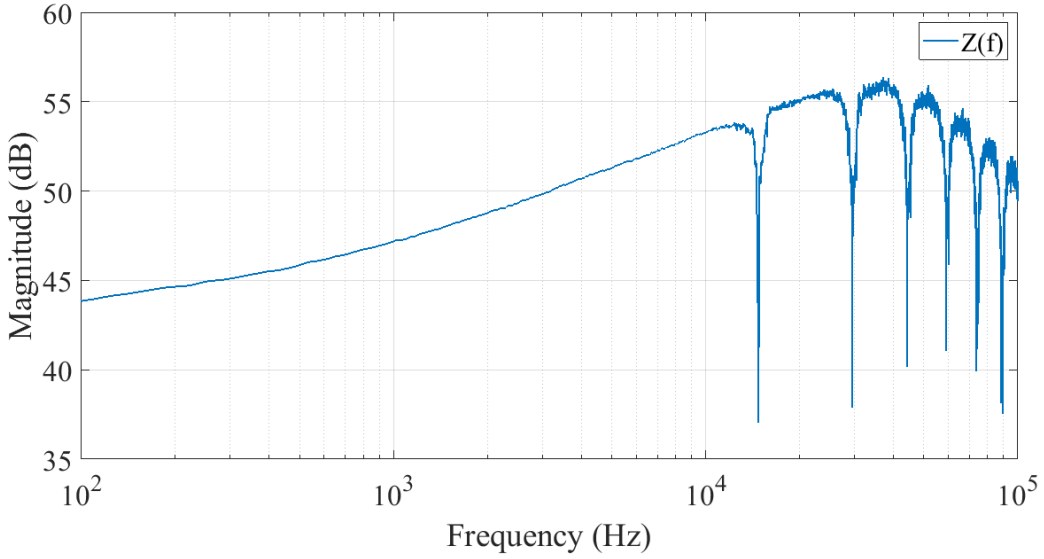



Figure F.11: Rheostat-Inductor Impedance Measurement

### F.2.3 Rheostat-Capacitive (RC) Load

**Voltage and Current PSD**

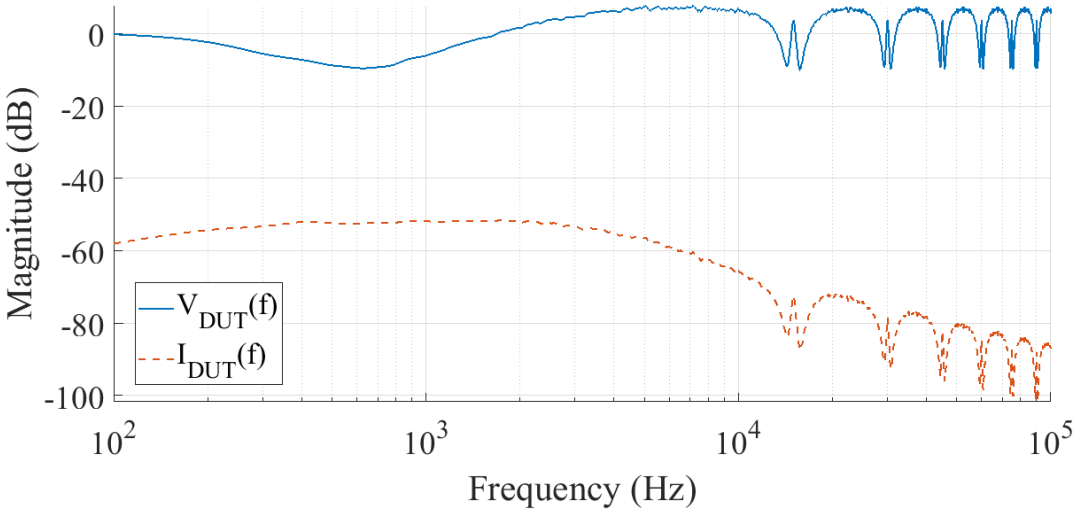



Figure F.12: Rheostat-Capacitor Simulation

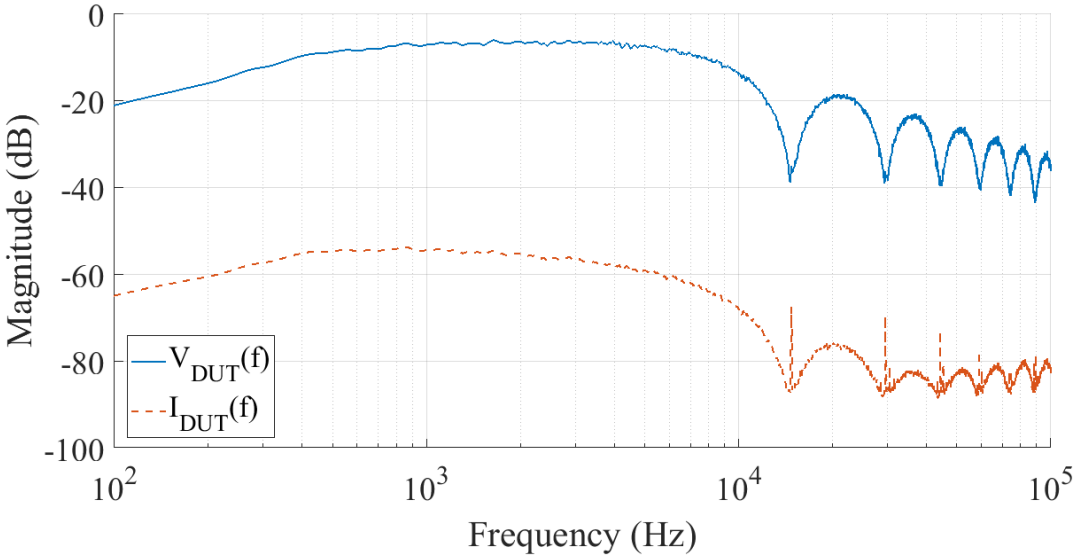



Figure F.13: Rheostat-Capacitor Measurement

### Impedance

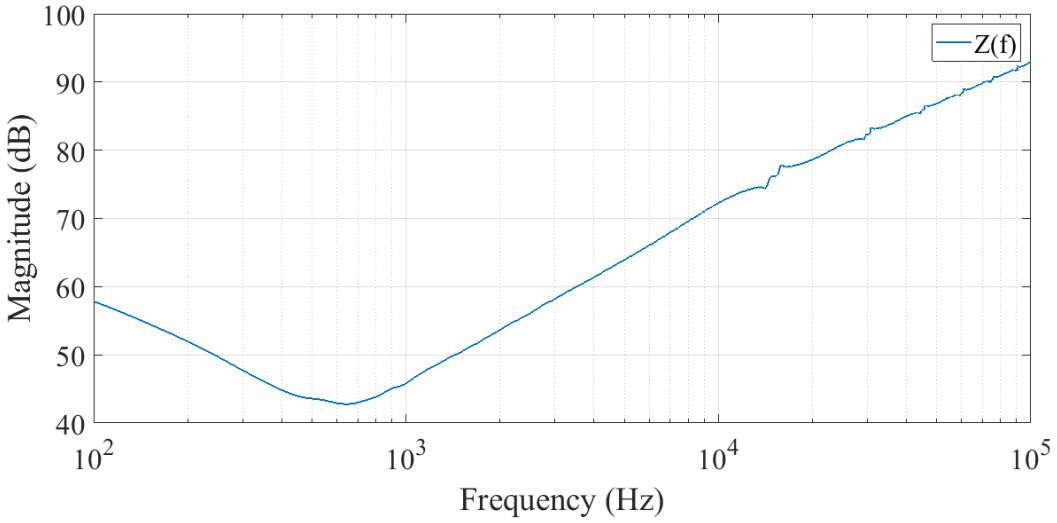



Figure F.14: Rheostat-Capacitor Impedance Simulation

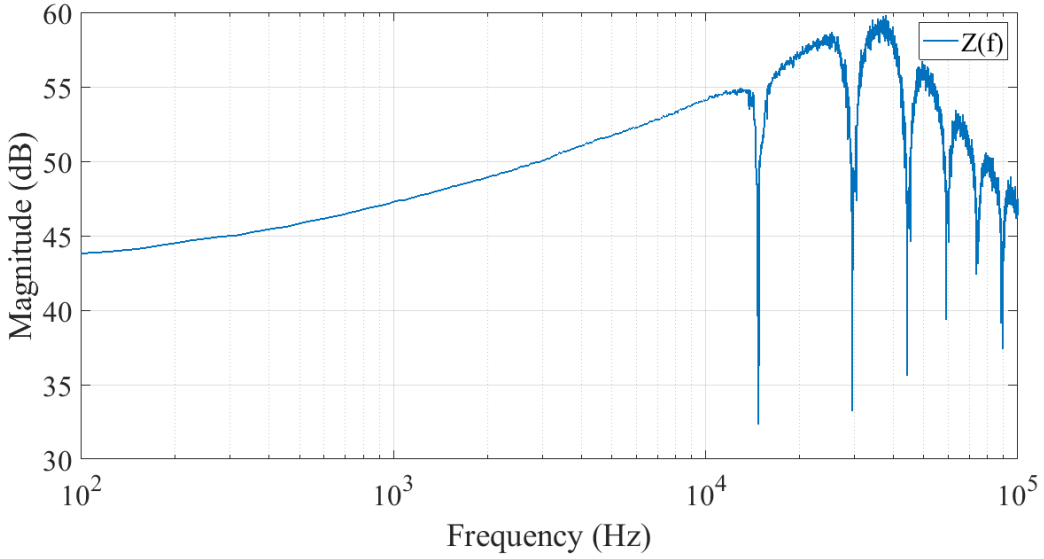



Figure F.15: Rheostat-Capacitor Impedance Measurement

### F.2.4 Rheostat-Inductive-Capacitive (RLC) Load

**Voltage and Current PSD**

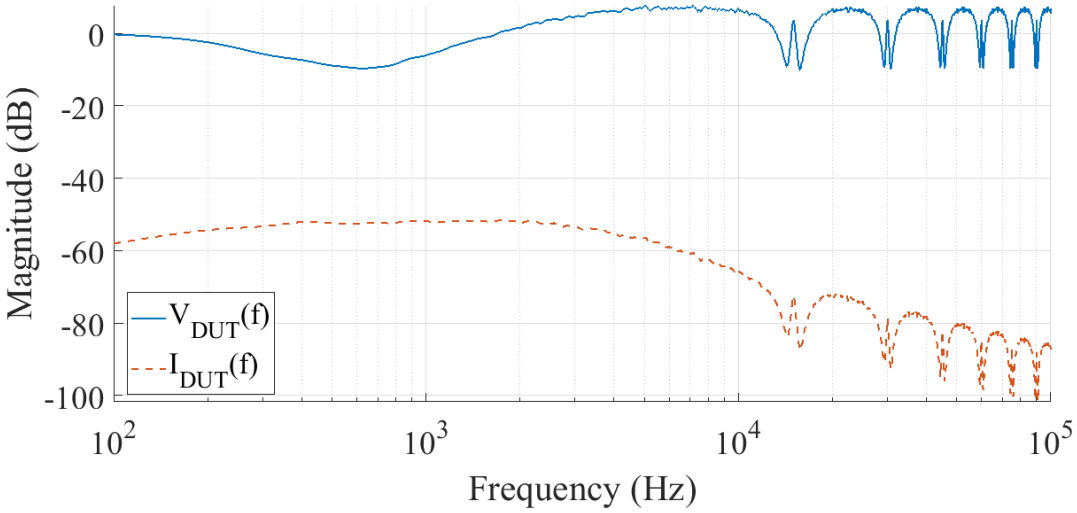



Figure F.16: Rheostat-Inductor-Capacitor Simulation

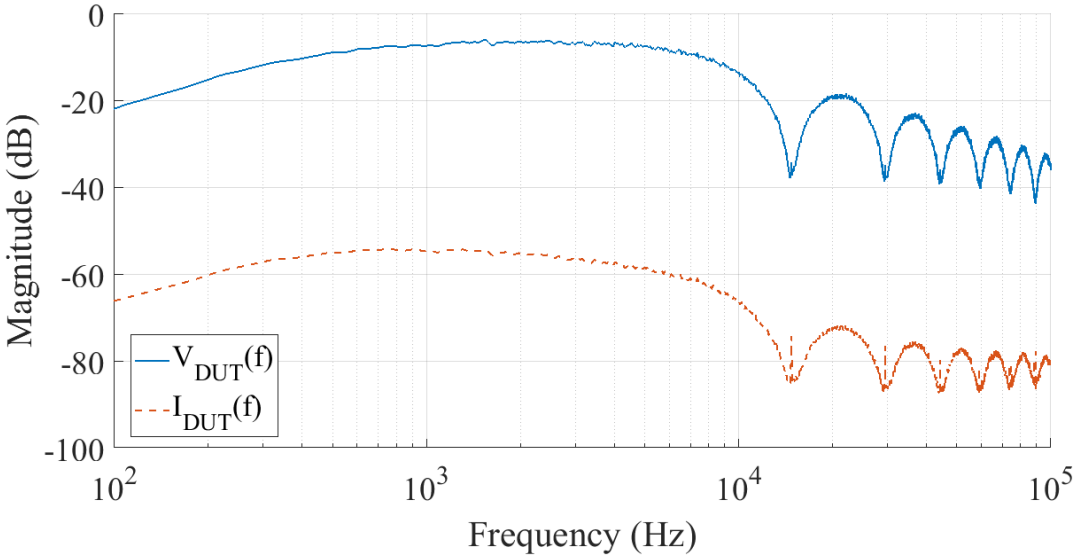



Figure F.17: Rheostat-Inductor-Capacitor Measurement

### Impedance

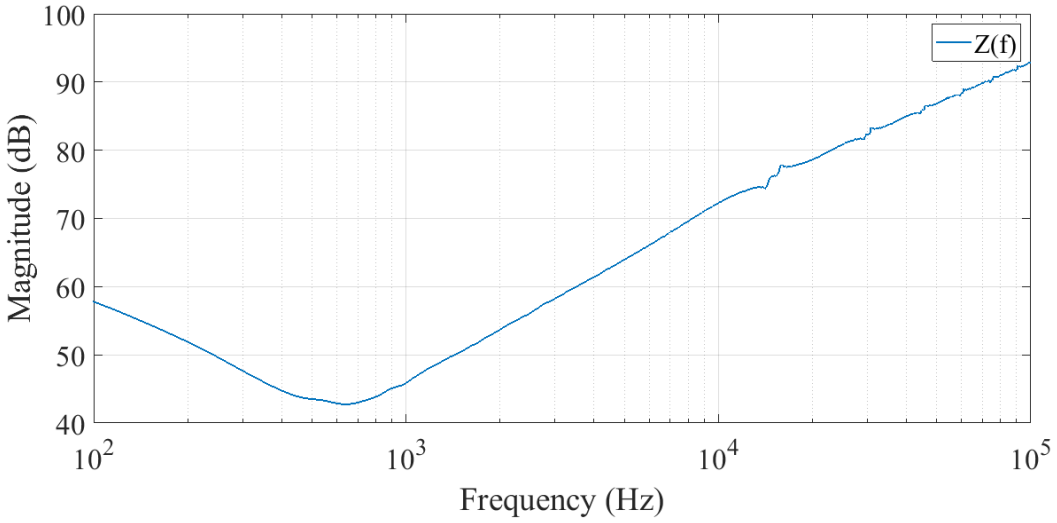



Figure F.18: Rheostat-Inductor-Capacitor Impedance Simulation

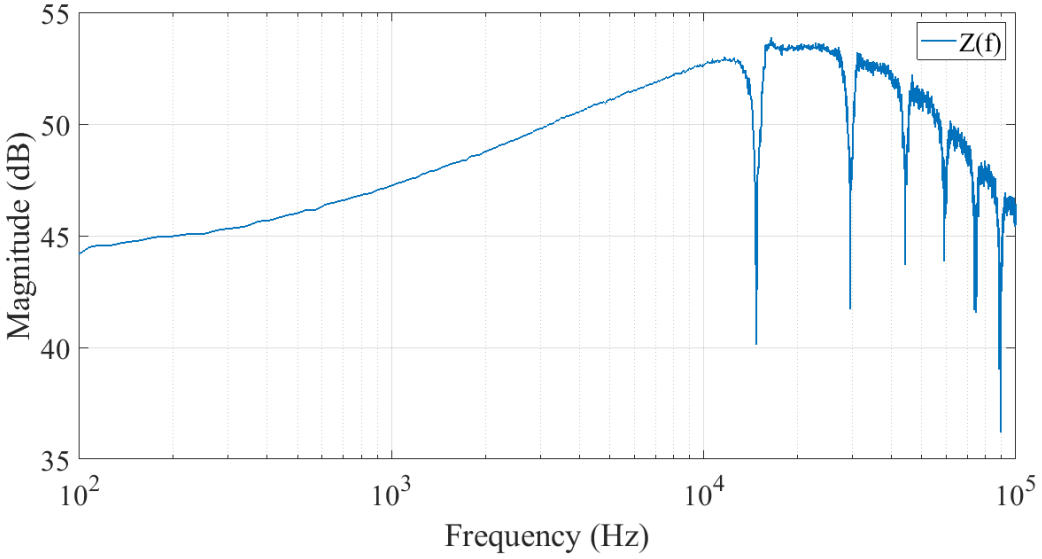



Figure F.19: Rheostat-Inductor-Capacitor Impedance Measurement

# Appendix G

# Arduino PRBS Generator Code

```
long toMicro = 1000000;

int CMD;
long frequency;
long duration;
double period;
int PRBSbits;
int periodLen;

// PRBS Polynomials
int polynomial[] = {
  0b1100, /* PRBS4: x^4 + x^3 + 1 */
  0b10100, /* PRBS5: x^5 + x^3 + 1 */
  0b110000, /* PRBS6: x^6 + x^5 + 1 */
  0b1100000, /* PRBS7: x^7 + x^6 + 1 */
  0b10111000, /* PRBS8: x^8 + x^6 + x^5 + x^4 + 1 */
  0b100010000, /* PRBS9: x^9 + x^5 + 1 */
  0b1001000000, /* PRBS10: x^10 + x^7 + 1 */
  0b10100000000, /* PRBS11: x^11 + x^9 + 1 */
  0b111000001000, /* PRBS12: x^12 + x^11 + x^10 + x^4 + 1 */
  0b1110010000000, /* PRBS13: x^13 + x^12 + x^11 + x^8 + 1 */
  0b11100000000010  /* PRBS14: x^14 + x^13 + x^12 + x^2 + 1 */
};

/* Start States
   Used as the seed for different PRBS bits. Any non-zero value
      will work
*/
int start_state[] = {
  0b1111,           // PRBS4
  0b11111,          // PRBS5
  0b111111,         // PRBS6
  0b1111111,        // PRBS7
```

```
33    0b11111111,       // PRBS8
34    0b111111111,      // PRBS9
35    0b1111111111,     // PRBS10
36    0b11111111111,    // PRBS11
37    0b111111111111,   // PRBS12
38    0b1111111111111,  // PRBS13
39    0b11111111111111  // PRBS14
40  };
41
42  /* Number of bits in a PRBSk period*/
43  int PRBS_periodLen[] = {
44    15,    // PRBS4
45    31,    // PRBS5
46    63,    // PRBS6
47    127,   // PRBS7
48    255,   // PRBS8
49    511,   // PRBS9
50    1023,  // PRBS10
51    2047,  // PRBS11
52    4095,  // PRBS12
53    8191,  // PRBS13
54    16383, // PRBS14
55  };
56
57  void setup() {
58    /* INITIALISE */
59
60    DDRB = B00001101; // !PRBS, PRBS, Gate Drive Disable
61    DDRC = B00010000; // Sample Frequency
62    DDRD = B01000000; // PRBS Clock
63    Serial.begin(1000000); // Baud rate for serial communication
64  }
65
66  void loop() {
67    /* MAIN LOOP */
68
69    while (Serial.available() > 0) { // While the input buffer is not
          empty
70
71      char bufferCMD;
72      char bufferFREQ[6];
73      char bufferPRBS[2];
74      char bufferPeriod[2];
75      char Duration[5];
76
77      bufferCMD = Serial.read();
78      CMD = bufferCMD;
```

```
79
80      if (CMD == 'y') {
81      // PRBS
82        String PRBS = Serial.readStringUntil(',');
83        PRBSbits = (PRBS.toInt());
84
85        // FREQUENCY
86        String FREQUENCY = Serial.readStringUntil(',');
87        frequency = atol(FREQUENCY.c_str());
88
89        // DURATION
90        String DURATION = Serial.readStringUntil('z');
91        duration = atol(DURATION.c_str());
92        bitshifter(PRBSbits,frequency,duration);
93
94      }
95    else {
96        delayMicroseconds(200);
97        SYS_pause();
98        delayMicroseconds(200);
99      }
100   }
101   SYS_pause();
102 }
103
104 void bitshifter( int PRBS_bits, long frequency, long duration) {
105
106   /*
107   PRBS_bits have been offset by 4 due to array indexing
108   */
109   int taps = polynomial[PRBS_bits - 4];
110   int lfsr = start_state[PRBS_bits - 4];
111   int loc_start = start_state[PRBS_bits - 4];
112   int PRBS_Period = PRBS_periodLen[PRBS_bits-4];
113
114   long tSamp = ((long)toMicro  / ((long)frequency * 2 ));
115   long tSamp2 = tSamp-1.5;
116
117   PORTD = B00000000;
118   PORTC = B00000000;
119
120   int lsb;
121   long tStart = micros();
122
123   long durMS = duration*1000;
124   long LStart = (long)tStart;
125
```

```
126   while ((micros() - LStart ) < durMS)
127   {
128     delayMicroseconds(tSamp2);
129     lsb = lfsr & 1; /* Get LSB (i.e., the output bit). */
130     lfsr >>= 1; /* Shift entire PRBs sequence */
131
132     if (lsb == 1) { /* Only apply toggle mask if output bit is 1.
          */
133       lfsr ^= taps; /* Apply toggle mask, value has 1 at bits
            corresponding to taps, 0 elsewhere. */
134      // ONES
135      PORTC ^= B00010000;
136      PORTB = B00000100;
137     }
138     else{
139      //ZEROS
140      PORTC ^= B00010000;
141      PORTB = B00001000;
142     }
143
144   if(lfsr==loc_start)
145   {
146     PORTD ^= B01000000;
147   }
148
149   if((Serial.available() > 0) && (Serial.read()=='x')){
150     SYS_pause();
151     break;
152     }
153
154   delayMicroseconds(tSamp2);
155   PORTC ^= B00010000;
156   }
157
158 }
159
160 void SYS_pause() {
161   delayMicroseconds(200);
162   PORTB = B00000001; // Open both switches using Gate Drive Disable
        Pin
163   delayMicroseconds(200);
164   PORTC = B00000000;
165 }
```

Listing G.1: SignalGenerator

# Appendix H

# Python Graphical User Interface Code

```
# Graphical User Interface
# Pseudo Random Binary Sequence generator
# Frequency – 10kHz,.....,100kHz
# Binary bits – 4...14
# Duration – As long Arduino Nano can take in long variable

import time
import serial
from tkinter import *


arduinoData = serial.Serial('COM7', baudrate=1000000)  # Initialise ....

global tempDuration

def CurrentFrequency(frequency):
    freqLookup = {'10 kHz': 10000, '15 kHz': 15000, '20 kHz': 20000, '25 kHz': 25000,
                  '30 kHz': 30000, '35 kHz': 35000, '40 kHz': 40000, '45 kHz': 45000,
                  '50 kHz': 50000, '55 kHz': 55000, '60 kHz': 60000, '65 kHz': 65000,
                  '70 kHz': 70000, '75 kHz': 75000, '80 kHz': 80000, '85 kHz': 85000,
                  '90 kHz': 90000, '95 kHz': 95000, '100 kHz': 100000}
    freq = freqLookup[frequency]
    return str(freq)


```

```
26 def CurrentPRBS(prbs):
27     prbsLookup = {'4': 4, '5': 5, '6': 6, '7': 7, '8': 8, '9': 9, '
           10': 10, '11': 11,
28             '12': 12, '13': 13, '14': 14}
29     prbs = prbsLookup[prbs]
30     return str(prbs)
31
32
33 def CurrentPeriod(period):
34     periodLookup = {'1': 1, '2': 2, '3': 3, '4': 4, '5': 5, '6': 6,
           '7': 7, '8': 8,
35             '9': 9, '10': 10}
36     period = periodLookup[period] + periodOffset
37     return str(period)
38
39
40 def tempDur():
41     global tempDuration
42     X = tempDuration
43     return X
44
45
46 class DemoClass:
47
48     def __init__(self):
49         self.window = Tk()
50         self.window.title("PRBS Generator")
51         self.window.geometry("400x275+500+50")
52         self.frame = Frame(self.window)
53         self.frame.grid(column=0, row=0, sticky=(N, W, E, S))
54         self.frame.columnconfigure(0, weight=1)
55         self.frame.rowconfigure(0, weight=1)
56         self.frame.pack(pady=25, padx=25)
57
58 # ============================================================
59         self.runB = Button(self.frame, activeforeground="green",
               text="Run Test",
60                            command=self.runTest)  #
61         self.runB.grid(row=7, column=3)
62
63         self.stopB = Button(self.frame, activeforeground="red",
               text="Stop Test",
64                            command=self.stopTest)  #
65         self.stopB.grid(row=7, column=1)
66
67         self.label1 = Label(self.frame, text="PRBS bits")  #
68         self.label1.grid(row=2, column=1)
```

```
69
70          self.label2 = Label(self.frame, text="Frequency")  #
71          self.label2.grid(row=2, column=2)
72
73          self.label3 = Label(self.frame, text="Duration (ms)")  #
74          self.label3.grid(row=2, column=3)
75
76          self.durButton = Button(self.frame, text='Set', command=
                self.processDuration)
77          self.durButton.grid(row=4, column=3)
78
79          self.CurrentSetDuration = Label(self.frame, text=" ")  #
80          self.CurrentSetDuration.grid(row=5, column=1, columnspan=3)
81
82          self.labelY = Label(self.frame, text=" ")  #
83          self.labelY.grid(row=6, column=1)
84
85  # ============================================================
86
87          # self.window.after(1000, self.timedFunction)
88          # self.window.mainloop()
89
90  # ============================================================
91          self.freqVar = StringVar(self.frame)
92          self.prbsVar = StringVar(self.frame)
93          self.periodsVar = StringVar(self.frame)
94
95          # Frequency options
96          frequencyChoices = ('10 kHz', '15 kHz', '20 kHz', '25 kHz',
97                              '30 kHz', '35 kHz', '40 kHz', '45 kHz',
98                              '50 kHz', '55 kHz', '60 kHz', '65 kHz',
99                              '70 kHz', '75 kHz', '80 kHz', '85 kHz',
100                             '90 kHz', '95 kHz', '100 kHz')
101         self.freqVar.set('10 kHz')  # set the default option
102
103         # PRBS bit options
104         PRBSchoices = ('4', '5', '6', '7', '8', '9', '10', '11', '
                12', '13',
105                  '14')
106         self.prbsVar.set('4')
107
108         periodChoices = ('1', '2', '3', '4', '5', '6', '7', '8', '9
                ', '10')
109         self.periodsVar.set('1')
110
111         popupMenu1 = OptionMenu(self.frame, self.prbsVar, *
                PRBSchoices)
```

```
112         popupMenu1.grid(row=3, column=1)
113
114         popupMenu2 = OptionMenu(self.frame, self.freqVar, *
                frequencyChoices)
115         popupMenu2.grid(row=3, column=2)
116
117         self.EntryField1 = Entry(self.frame, width=8)
118         self.EntryField1.grid(row=3, column=3)
119 # ============================================================
120
121         tempDur = self.EntryField1.get()
122         durButton = Button(self.frame, text='Set', command=self.
                processDuration)
123         durButton.grid(row=4, column=3)
124 # ============================================================
125
126         self.DateLabel = Label(self.frame, text="")
127         self.DateLabel.grid(row=0, column=1, columnspan=3)
128         self.ClockLabel = Label(self.frame, text="")
129         self.ClockLabel.grid(row=1, column=1, columnspan=3)
130
131         self.update_clock()
132         self.window.mainloop()
133
134 # ====================Clock==================================
135
136     def update_clock(self):
137         now = time.strftime("%H:%M:%S")
138         date = time.strftime("%A %d %B %Y")
139         self.ClockLabel.configure(text="Time: " + now)
140         self.DateLabel.configure(text="Date: " + date)
141         self.window.after(1000, self.update_clock)
142
143     # ============================================================
144     def processDuration(self):
145         print('Testing Duration Entry')
146         global tempDuration
147         tempDuration = self.EntryField1.get()
148         self.CurrentSetDuration.configure(text="Set Duration is: "
                + tempDuration + " ms")
149         print('Temp Duration = ', tempDuration)
150
151
152     def stopTest(self):
153         stopTest = 'x'
154         arduinoData.write(stopTest.encode())
155         print('Stop Test', stopTest)
```

```
156
157     def runTest(self):
158         comma = str(',')
159         beginTrans = 'y'  # Indicates that a test is about commence
                , for Arduino
160         endTrans = 'z'  # Indicates end of transmission
161         sampFreq = CurrentFrequency(self.freqVar.get())  # Sampling
                Frequency
162         prbsL = CurrentPRBS(self.prbsVar.get())  # PRBS length
163         Duration = tempDur()
164
165         transmitData = beginTrans + prbsL + comma + sampFreq +
                comma + Duration + endTrans
166         arduinoData.write(transmitData.encode())
167         print(transmitData)
168
169     def timedFunction(self):
170     #Debugging function
171         print("I'm in a function that keeps rescheduling itself to
                be executed after 1 second!")
172         self.window.after(1000, self.timedFunction)
173
174 instance = DemoClass()
```

Listing H.1: PRBS GUI code

# Appendix I

# MATLAB Scripts

```
frequency = 15000;       % PRBS frequency
Tclock = 1/frequency;    % Clock frequency
impulseLen = 1000e-6;    % Longest PRBS run (Rounded up)
PRBSrun = 14;            % PRBS length design

syms R L C tau1 tau2 s1 s2; % Declaring symbolic variables
X = 5; % Set the number of time constants for tau2 in eqn3

tau1 = 0.01*Tclock; % Time constant is equal to 1% of period
% Increasing tau1 increases the inductor size
% tau2 is set by eqn3 due to constraints
% tau2 is left as an unknown to not overconstrain the equations

C = 2e-6; % Setting a capacitor
% L or C  can be fixed in order to calculate other parameters

s1 = 1/tau1;
s2 = 1/tau2;

% The roots of a series RLC circuit. (CURRENT)
eqn1 = s1 == (-R/(2 * L)) + sqrt(((C * R.^2) -(4 * L) )/(4 * C * L
    .^2)) ;
eqn2 = s2 == (-R/(2 * L)) - sqrt(((C * R.^2) -(4 * L) )/(4 * C * L
    .^2)) ;

% Longest run of a PRBS signal. It has to rise and fall in that
    duration
% eqn3 = (5*tau1)+ (X*tau2) == impulseLen;
eqn3 = (5*tau1)+ (X*tau2) == (PRBSrun*Tclock);

% Solving system of equations
sol = solve([eqn1, eqn2, eqn3],[R L tau2]);

```

```
Resistor = sol.R;
Inductor = sol.L;
t2 = double(sol.tau2);

% The solutions
resistorR = double(abs(sol.R)); % Ohm - RLC Resitor
inductorL = double(abs(sol.L)); % Henry - RLC Inductor
%capacitorC = abs(sol.C); % Farad - RLC Capacitor
capacitorC = C; % Farad - RLC Capacitor
```

Listing I.1: RLC Optimiser script

```
function psdScope(V,I)

ft = 200:200000;

vLoad = V;
iLoad = I;

sampFreq = 625000;
nfft = length(vLoad);

[psdV,f] = pwelch(vLoad,hamming(round(nfft/100)),round(nfft/1000),
    round(length(vLoad)),sampFreq,'psd');
[psdI,f] = pwelch(iLoad,hamming(round(nfft/100)),round(nfft/1000),
    round(length(iLoad)),sampFreq,'psd');
psdZ = sqrt(psdV./psdI);

fX = figure;
fY = figure;

figure(fX)
hold on
grid on

semilogx(f(ft),10*log10(psdV(ft)),'LineWidth',1.5)
semilogx(f(ft),10*log10(psdI(ft)),'--','LineWidth',1.5)
set(gca, 'XScale', 'log')
set(gcf,'color','w');
ax = gca;
ax.FontSize = 28;
ax.FontName = 'Serif';

legend('V_{DUT}(f)','I_{DUT}(f)')
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')
```

```
33
34 hold off
35
36 figure(fY)
37 semilogx(f(ft),20*log10(psdZ(ft)),'LineWidth',1.5)
38 set(gcf,'color','w');
39 ax = gca;
40 ax.FontSize = 25;
41 ax.FontName = 'Serif';
42 legend('Z(f)')
43 xlabel('Frequency (Hz)')
44 ylabel('Magnitude (dB)')
45 grid on
```

Listing I.2: Power Spectral Density. Voltage and Current

```
1  function psdScopeBI(PRBSin,PRISin)
2
3  ft = 200:200000;
4
5  PRBs = PRBSin;
6  PRIs = PRISin;
7
8  sampFreq = 625000;
9  nfft = length(PRBs);
10
11 [psdB,f] = pwelch(PRBs,hamming(round(nfft/100)),round(nfft/1000),
       round(length(PRBs)),sampFreq,'psd');
12 [psdI,f] = pwelch(PRIs,hamming(round(nfft/100)),round(nfft/1000),
       round(length(PRIs)),sampFreq,'psd');
13
14 fX = figure;
15
16 figure(fX)
17 hold on
18 grid on
19
20 semilogx(f(ft),10*log10(psdB(ft)),'LineWidth',1.5)
21 semilogx(f(ft),10*log10(psdI(ft)),'--','LineWidth',1.5)
22 set(gca, 'XScale', 'log')
23 set(gcf,'color','w');
24 ax = gca;
25 ax.FontSize = 28;
26 ax.FontName = 'Serif';
27
28 legend('PRBS(f)','PRIS(f)')
29 xlabel('Frequency (Hz)')
```

```
30 ylabel('Magnitude (dB)')
31
32 hold off
```

Listing I.3: Power Spectral Density. PRBS and PRIS